COURSE OUTLINE BRIEFS

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SARGODHA UNIVERSITY
Pathway to Progress

FACULTY OF SCIENCES

OVERVIEW

The Department of Computer Sciences and Information Technology was established at the Ibn-ul- Haytham Block in January 2003, where it started with the first batch of Bachelor of Computer Science (BSCS) and Master of Science (MCS) in March 2003. It encompasses Computer Science (CS), Information Technology (IT) and Software Engineering (SE).

The Department is committed to provide a high quality education for solving real world problems and leadership skills to its undergraduates and graduates in theoretical, practical and applied foundations of computer science.

To meet the ever-growing needs in field of CS and IT, the Department offers PhD, MSCS, MSIT, MSc (IT), BS (CS), BS (IT) and BS (SE) programs. The Department has highly qualified faculty including four PhD and 21 MS/MPhil qualified members.

The faculty is carrying out research along with students across a broad range of subjects within CS&IT including Machine Learning, Data Science, Fog Computing, IOT, Natural Language Processing, Software Engineering, Algorithms and Computing.

Academic Programs Offered

1. BS Computer Science
2. BS Information Technology
3. BS Software Engineering
4. MS Compute Science
5. MS Information Technology
6. PhD Computer Science

BS Computer Science

Eligibility: At least 50% marks in Intermediate (HSSC) examination with Mathematics or equivalent qualification with Mathematics certified by IBCC.
Duration: 04 Year Program (08 Semesters)
Degree Requirements: 130 Credit Hours

Semester I

Course Code	Course Title	Credit Hours	Pre-requisite
ICTC-5201	Introduction to ICT	3(2+1)	
CMPC-5201	Programming Fundamentals	4(3+1)	
ENGL-5101	Functional English	3(3+0)	
MATH-5101	Calculus & Analytical Geometry	3(3+0)	
PHYS-5101	Applied Physics	3(3+0)	
URCI-5105	Islamic Studies	2(2+0)	

Semester II

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-5201	Digital Logic and Design	4(3+1)	
CMPC-5202	Object Oriented Programming	4(3+1)	Programming Fundamentals
MATH-5102	Probability & Statistics	3(3+0)	
CMPC-5101	Discrete Structure	3(3+0)	
ENGL-5102	Communication Skills	3(3+0)	Functional English

Semester III

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-5202	Computer Organization & Assembly Language	4(3+1)	
CMPC-5203	Data Structures & Algorithms	4(3+1)	Object Oriented Programming
ENGL-5103	Technical & Report Writing	3(3+0)	Communication Skills
URCP-5106	Pakistan Studies	2(2+0)	

CMPC-5205	Computer Networks	4(3+1)	

Semester IV

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-5101	Design & Analysis of Algorithms	3(3+0)	Data Structures & Algorithms
CMPC-5204	Database Systems	4(3+1)	
MATH-5103	Linear Algebra	3(3+0)	
CMPC-5103	Software Engineering	3(3+0)	
xxxx-4xxx	University Elective-II	3(3+0)	

Semester V

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-6101	Theory of Automata	3(3+0)	
CMPC-6201	Operating Systems	4(3+1)	Data Structures & Algorithms
CSEC-6xxx	CS Elective-I	3(3+0)	
CSSC-6xxx	CS Supporting-I	3(3+0)	
CSSC-6xxx	CS Supporting-II	3(3+0)	

Semester VI

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-6201	Artificial Intelligence	3(3+1	Discrete Structures
CSCC-6103	Compiler Construction	3(3+0)	Theory of Automata
CSCC-6102	Parallel & Distributed Computing	3(3+0)	
CMPC-6102	Information Security	3(3+0)	
CSEC-6xxx	CS Elective-II	3(3+0)	

Semester VII

CMPC-6701	Capstone	3(0+3)	Pre-requisite
CSEC-6xxx	CS Elective-III	3(3+0)	
xxxx-6xxx	University Elective-III	3(3+0)	
CSEC-6xxx	CS Elective-IV	3(3+0)	

Semester VIII

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-6702	Capstone	3(0+3)	
CSSC-6101	Professional Practice	3(3+0)	
CSEC-6xxx	CS Elective-V	3(3+0)	
xxxx-6xxx	University Elective-IV	3(3+0)	

BS Information Technology

Eligibility: At least 50% marks in Intermediate (HSSC) examination with Mathematics or equivalent qualification with Mathematics certified by IBCC.
Duration: 04 Year Program (08 Semesters)
Degree Requirements: 130 Credit Hours

Semester I

Course Code	Course Title	Credit Hours	Pre-requisite
ICTC-5201	Introduction To ICT	3(2+1)	
CMPC-5201	Programming Fundamentals	4(3+1)	
ENGL-5101	English Composition & Comprehension	3(3+0)	
MATH-5101	Calculus & Analytical Geometry	3(3+0)	
PHYS-5101	Applied Physics/Quantum Computing	3(3+0)	

Semester II

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5202	Object Oriented Programming	4(3+1)	Programming Fundamentals
ENGL-5102	Communication & Presentation Skills	3(3+0)	
ITSC-5xxx	IT Supporting Course – I	3(3+0)	
MATH-5102	Probability & Statistics	3(3+0)	
xxxx-5xxx	University Elective – 1	3(3+0)	
xxxx-5xxx	University Elective – 2	3(3+0)	

Semester III

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5203	Data Structures & Algorithms	4(3+1)	Object Oriented Programming
CMPC-5101	Discrete Structures	3(3+0)	
ITSCC-5101	Professional Practices	3(3+0)	
ITSC-5xxx	IT Supporting Course – II	3(3+0)	
MATH-5103	Linear Algebra	3(3+0)	

Semester IV

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5204	Operating Systems	4(3+1)	Data Structures & Algorithms
CMPC-5102	Software Engineering	3(3+0)	

CMPC-5205	Computer Networks	4(3+1)	
ITCC-5101	IT project Management	3(3+0)	
xxxx-5xxx	University Elective – 3	3(3+0)	

Semester V

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-6201	Database Systems	4(3+1)	Data Structures and Algorithms
CMPC-6101	Information Security	3(3+0)	
ITSC-6xxx	IT Supporting Course – III	3(3+0)	
ITCC-6202	System and Network Administration	4(3+1)	Operating Systems
ITEC-6xxx	IT Elective – 1	3(3+0)	

Semester VI

Course Code	Course Title	Credit Hours	Pre-requisite
ITCC-6103	Web Systems and Technologies	3(3+0)	
xxxx-6xxx	University Elective – 4	3(3+0)	
ITEC-6xxx	IT Elective – 2	3(3+0)	
ITEC-6xxx	IT Elective – 3	3(3+0)	
ENGL-6101	Technical & Business Writing	3(3+0)	

Semester VII

ITCC-6203	Virtual Systems and Services	4(3+1)	Pre-requisite
CMPC-6701	Capstone project-I	3(0+3)	
ITCC-6102	IT infrastructure	3(3+0)	
ITEC-6xxx	IT Elective – 4	3(3+0)	
URCP-5106	Pakistan Studies	2(2+0)	

Semester VIII

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-6702	Capstone Project-II	3(0+3)	
ITCC-6101	Cyber Security	3(3+0)	
ITEC-6xxx	IT Elective – 5	3(3+0)	
ITCC-6201	Database Administration and Management	4(3+1)	
URCI-5105	Islamic Studies/ Ethics	2(2+0)	

BS Software Engineering

Eligibility: At least 50% marks in Intermediate (HSSC) examination with Mathematics or equivalent qualification with Mathematics certified by IBCC.
Duration: 04 Year Program (08 Semesters)
Degree Requirements: 130 Credit Hours

Semester I

Course Code	Course Title	Credit Hours	Pre-requisite
ICTC-5201	Introduction to Information & Communication Technologies	3(2+1)	
CMPC-5201	Programming Fundamentals	4(3+1)	
ENGL-5101	English Composition & Comprehension	3(3+0)	
MATH-5101	Calculus & Analytical Geometry	3(3+0)	
URCP-5106	Pakistan Studies	2(2+0)	
PHYS-5101	Applied Physics	3(3+0)	

Semester II

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5202	Object Oriented Programming	4(3+1)	Programming Fundamentals
ENGL-5102	Communication & Presentation Skills	3(3+0)	English Composition & Comprehension
CMPC-5101	Discrete Structures	3(3+0)	
CMPC-5102	Software Engineering	3(3+0)	
URCI-5105	Islamic Studies	2(2+0)	
xxxx-5xxx	University Elective – I	3(3+0)	

Semester III

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5203	Data Structures & Algorithms	4(3+1)	Object Oriented Programming
SECC-5201	Software Requirement Engineering	3(3+0)	Software Engineering
SECC-5202	Human Computer Interaction	3(3+0)	Software Engineering
MATH-5102	Linear Algebra	3(3+0)	
xxxx-5xxx	University Elective-II	3(3+0)	

Semester IV

Course Code	Course Title	Credit Hours	Pre-requisite
CMPC-5204	Operating Systems	4(3+1)	Data Structures & Algorithms
CMPC-5205	Database Systems	4(3+1)	Data Structures & Algorithms

SECC-202	Software Design and Architecture	3(2+1)	Software Engineering
MATH-5103	Probability and Statistics	3(3+0)	
xxxx-5xxx	University Elective – III	3(3+0)	

Semester V

Course Code	Course Title	Credit Hours	Pre-requisite
SECC-6201	Software Construction and Development	3(2+1)	Software Design and Architecture
CMPC-6201	Computer Networks	4(3+1)	
ENGL-6101	Technical and Report Writing	3(3+0)	Communication & Presentation Skills
SESC-6xxx	SE Supporting –I	3(3+0)	
SESC-6xxx	SE Supporting – II	3(3+0)	

Semester VI

Course Code	Course Title	Credit Hours	Pre-requisite
SECC-6101	Software Quality Engineering	3(3+0)	Software Engineering
CMPC-6101	Information Security	3(3+0)	
SESC-6101	Professional Practice	3(3+0)	
SECC-6102	Web Engineering	3(3+0)	
SEEC-6xxx	SE Elective – I	3(3+0)	
SESC-6xxx	SE Supporting – III	3(3+0)	

Semester VII

Course Code	Course Title	Credit Hours	Pre-requisite
SECC-6103	Software Project Management	3(3+0)	Software Engineering
SECC-6104	Software Re-Engineering	3(3+0)	
SEEC-6xxx	SE Elective –II	3(3+0)	
SEEC-6xxx	SE Elective – III	3(3+0)	
CMPC-6701	Capstone – I	3(0+3)	

Semester VIII

Course Code	Course Title	Credit Hours	Pre-requisite
SEEC-6xxx	SE Elective – IV	3(0+3)	
SEEC-6xxx	SE Elective – V	3(0+3)	
CMPC-6702	Capstone – II	3(0+3)	
xxxx-6xxx	University Elective – IV	3(0+3)	

MS Computer Science

Eligibility: 16-years of education with 2.5 CGPA or equivalent in science/engineering discipline preferably with 4 years degree program of BS (SE/CS/IT/EE/CE) or equivalent from HEC recognized university or degree awarding institute.
Duration: 02 Year Program (04 Semesters)
Degree Requirements: 30 Credit Hours

Semester I

Course Code	Course Title	Credit Hours	Pre-requisite
CSCC-7401	Advanced Theory of Computation	3(3+0)	
CSCC-7402	Advanced Algorithm Analysis	3(3+0)	
CSCC-7403	Advanced Operating Systems	3(3+0)	

Semester II

Course Code	Course Title	Credit Hours	
CSCC-7404	Advanced Computer Architecture	3(3+0)	
	Elective I	3(3+0)	
	Elective II	3(3+0)	

Semester III

Course Code	Course Title	Credit Hours	
CSCC-7405	Thesis-I	3(0+3)	
	Elective III	3(3+0)	
	Elective IV	3(3+0)	

Semester IV

Course Code	Course Title	Credit Hours	
CSCC-7405	Thesis-II	3(0+3)	

List of Elective Courses

Course Code	Course Title	Cr. Hrs.	Pre. Req.
ITCC-7401	Advanced Topics in Wireless Networks	3(3+0)	Computer Networks
CSEC-7401	Advanced Artificial Intelligence	3 (3+0)	Artificial Intelligence

CSEC-7402	Advanced Human Computer Interaction	3(3+0)	Human Computer Interaction
CSEC-7403	Advanced Web development	3(3+0)	Web system and technology
CSEC-7404	Advanced Web Engineering	3 (3+0)	None
CSEC-7405	Agent Based Modeling	3(3+0)	None
CSEC-7406	Big Data Analysis	3(3+0)	None
CSEC-7407	Bioinformatics	3(3+0)	None
ITEC-7401	Business Process Management Trends & Technologies	3 (3+0)	None
CSEC-7408	Cloud Computing	3(3+0)	Computer Networks
CSEC-7409	Computational Intelligence	3(3+0)	None
CSEC-7410	Computer Vision		Linear Algebra
CSEC-7411	Digital Image Processing	3(3+0)	None
CSEC-7412	Distributed Computing	3(3+0)	DBMS
ITEC-7402	Distributed Database Systems	3(3+0)	DBMS
CSEC-7413	Distributed System	3(3+0)	OS
ITEC-7403	Enterprise Data Center Design and Methodology	3 (3+0)	None
CSEC-7414	Expert Systems	3(3+0)	None
CSEC-7415	Formal Methods	3(3+0)	None
CSEC-7416	Functional Programming	3(3+0)	None
CSEC-7417	Fuzzy Logic	3(3+0)	None
CSEC-7418	Game Theory	3(3+0)	Linear Algebra
CSEC-7419	Grammar Engineering	4(3+1)	None
CSEC-7420	Graph Theory & Algorithms	3(3+0)	Design and Analysis of Algorithms

CSEC-7421	Information Retrieval	3(3+0)	None
ITEC-7404	Information Security	3 (3+0)	None
CSEC-7422	Information Theory	3(3+0)	Probability & Statistics
CSEC-7423	Intelligent Multi Agent systems	3(3+0)	None
CSEC-7424	Intelligent Agents Technology	3(3+0)	None
ITEC-7405	Knowledge Management Systems & Technologies	3 (3+0)	None
CSEC-7425	Machine Learning	3(3+0)	None
ITEC-7406	Multimedia Databases	3(3+0)	None
CSEC-7426	Multimedia Systems Theory & Principles	3(3+0)	None
CSEC-7427	Natural Language Processing	3(3+0)	None
CSEC-7428	Parallel Processing: Algorithms and Architectures-I	3(3+0)	None
CSEC-7429	Parallel Processing: Algorithms and Architectures-II	3(3+0)	Parallel Processing: Algorithms and Architectures-I
SEEC-7401	Reliability Engineering	3(3+0)	Software Engineering
SEEC-7402	Research Methodology	3(3+0)	None
ITEC-7407	Service-Oriented Architecture	3(3+0)	None
CSEC-7430	Soft Computing	3(3+0)	None
SEEC-7403	Software Configuration Management	3(3+0)	Software Engineering
SEEC-7404	Software Measurement and Metrics	3(3+0)	Software Engineering
SEEC-7405	Software Risk Management	3(3+0)	Software Engineering
CSEC-7431	Speech Recognition and Synthesis	3(3+0)	None

CSEC-7432	Stochastic Processes	3(3+0)	Probability & Statistics
CSEC-7433	Stochastic Systems	3(3+0)	Probability & Statistics
ITEC-7408	Telecom Management	3 (3+0)	Computer Networks
ITEC-7409	Telecommunication Systems	3 (3+0)	Computer Networks
CSEC-7435	Text Processing and Data Analysis	4(3+1)	None
ITEC-7410	Virtualization in Cloud Computing	3 (3+0)	None
ITCC-7402	Information Privacy and Security	3 (3+0)	None
ITCC-7403	Cryptography	3 (3+0)	None
ITCC-7404	Advanced Database Management Systems	3 (3+0)	None
SECC-7406	Advanced Requirements Engineering	3(3+0)	None
SECC-7407	Advanced Software System Architecture	3(3+0)	None
SECC-7408	Software Testing and Quality Assurance	3(3+0)	None
SECC-7409	Advance Software Project Management	3(3+0)	None
CSEC-7436	Swarm Intelligence	3(3+0)	None
CSEC-7437	Evolutionary Computation	3(3+0)	None
CSEC-7438	Fog Computing	3(3+0)	None
CSEC-7439	Special Topics in Natural Language Engineering	3(3+0)	None
CSEC-7440	Advance Natural Language Engineering	3(3+0)	None

MS Information Technology

Eligibility: 16-years of education with 2.5 CGPA or equivalent in science/engineering discipline preferably with 4 years degree program of BS (SE/CS/IT/EE/CE) or equivalent from HEC recognized university or degree awarding institute.
Duration: 02 Year Program (04 Semesters)
Degree Requirements: 30 Credit Hours

Semester I

Course Code	Course Title	Credit Hours	
ITCC-7401	Advanced Topics in Wireless Networks	3(3+0)	
ITCC-7402	Information Privacy and Security	3(3+0)	
ITCC-7403	Cryptography	3(3+0)	

Semester II

Course Code	Course Title	Credit Hours	
ITCC-7404	Advanced Database Management System	3(3+0)	
	Elective I	3(3+0)	
	Elective II	3(3+0)	

Semester III

Course Code	Course Title	Credit Hours	
	Elective III	3(0+3)	
	Elective V	3(3+0)	
ITCC-7405	Thesis-I	3(3+0)	

Semester IV

Course Code	Course Title	Credit Hours	
ITCC-7405	Thesis-II	3(3+0)	

List of Elective Courses

Course Code	Course Title	Cr. Hrs.	Pre. Req.
ITEC-7401	Business Process Management Trends & Technologies	3(3+0)	None
ITEC-7402	Distributed Database Systems	3(3+0)	DBMS
ITEC-7403	Enterprise Data Center Design and Methodology	3(3+0)	None
ITEC-7404	Information Security	3(3+0)	None
ITEC-7405	Knowledge Management Systems & Technologies	3 (3+0)	None
ITEC-7406	Multimedia Databases	3(3+0)	None

ITEC-7407	Service-Oriented Architecture	3(3+0)	None
ITEC-7408	Telecom Management	3 (3+0)	Computer Networks
ITEC-7409	Telecommunication Systems	3 (3+0)	Computer Networks
ITEC-7410	Virtualization in Cloud Computing	3(3+0)	None
CSEC-7401	Advanced Artificial Intelligence	3 (3+0)	Artificial Intelligence
CSEC-7402	Advanced Human Computer Interaction	3(3+0)	Human Computer Interaction
CSEC-7403	Advanced Web development	3(3+0)	Web system and technology
CSEC-7404	Advanced Web Engineering	3 (3+0)	None
CSEC-7405	Agent Based Modeling	3(3+0)	None
CSEC-7406	Big Data Analysis	3(3+0)	None
CSEC-7407	Bioinformatics	3(3+0)	None
CSEC-7408	Cloud Computing	3(3+0)	Computer Networks
CSEC-7409	Computational Intelligence	3(3+0)	None
CSEC-7410	Computer Vision		Linear Algebra
CSEC-7411	Digital Image Processing	3(3+0)	None
CSEC-7412	Distributed Computing	3(3+0)	DBMS
CSEC-7413	Distributed System	3(3+0)	OS
CSEC-7414	Expert Systems	3(3+0)	None
CSEC-7415	Formal Methods	3(3+0)	None
CSEC-7416	Functional Programming	3(3+0)	None
CSEC-7417	Fuzzy Logic	3(3+0)	None
CSEC-7418	Game Theory	3(3+0)	Linear Algebra

CSEC-7419	Grammar Engineering	3(3+0)	None
CSEC-7420	Graph Theory & Algorithms	3(3+0)	Design and Analysis of Algorithms
CSEC-7421	Information Retrieval	3(3+0)	None
CSEC-7422	Information Theory	3(3+0)	Probability & Statistics
CSEC-7423	Intelligent Multi Agent systems	3(3+0)	None
CSEC-7424	Intelligent Agents Technology	3(3+0)	None
CSEC-7425	Machine Learning	3(3+0)	None
CSEC-7426	Multimedia Systems Theory & Principles	3(3+0)	None
CSEC-7427	Natural Language Processing	3(3+0)	None
CSEC-7428	Parallel Processing: Algorithms and Architectures-I	3(3+0)	None
CSEC-7429	Parallel Processing: Algorithms and Architectures-II	3(3+0)	Parallel Processing: Algorithms and Architectures-I
CSEC-7430	Soft Computing	3(3+0)	None
CSEC-7431	Speech Recognition and Synthesis	3(3+0)	None
CSEC-7432	Stochastic Processes	3(3+0)	Probability & Statistics
CSEC-7433	Stochastic Systems	3(3+0)	Probability & Statistics
CSEC-7434	Advanced Theory of Computation	3(3+0)	None
CSEC-7435	Advanced Operating Systems	3(3+0)	Operating Systems
CSEC-7436	Swarm Intelligence	3(3+0)	None
CSEC-7437	Evolutionary Computation	3(3+0)	None
CSEC-7438	Fog Computing	3(3+0)	None

CSEC-7439	Advanced Computer Architecture	3(3+0)	Computer Architecture
CSEC-7440	Advance Natural Language Engineering	3(3+0)	None
CSEC-7441	Special Topics in Natural Language Engine	3(3+0)	None
CSEC-7442	Text Processing and Data Analysis	3(3+0)	None
CSEC-0837	Text Processing & Data Analysis	3(3+0)	None
CSEC-7444	Advanced Algorithm Analysis		Analysis of Algorithms
SEEC-7401	Reliability Engineering	3(3+0)	Software Engineering
SEEC-7402	Research Methodology	3(3+0)	None
SEEC-7403	Software Configuration Management	3(3+0)	Software Engineering
SEEC-7404	Software Measurement and Metrics	3(3+0)	Software Engineering
SEEC-7405	Software Risk Management	3(3+0)	Software Engineering
SECC-7406	Advanced Requirements Engineering	3(3+0)	None
SECC-7407	Advanced Software System Architecture	3(3+0)	None
SECC-7408	Software Testing and Quality Assurance	3(3+0)	None
SECC-7409	Advance Software Project Management	3(3+0)	None

PhD Computer Science

Eligibility: minimum CGPA 3.0 (out of 4.0 in the Semester System) or First Division (in the Annual System) in M.Phil./M. S/Equivalent degree in Computer Science or related disciplines.
Duration: 03 Year Program (06 Semesters)
Degree Requirements: 24 Credit Hours

List of Elective Courses

Course Code	Course Title	Credit Hours
CSEC-8401	Special Topics in Natural Language Engineering	3(3+0)
CSEC-8402	Special topics in Grammar Engineering	3(3+0)
CSEC-8403	Advanced Topics in Formal Methods	3(3+0)
CSEC-8404	Special Topics in Cluster & Grid Computing	3(3+0)
CSEC-8405	Advanced topics in Parallel Processing	3(3+0)
CSEC-8406	Special Topics in Distributed Computing	3(3+0)
CSEC-8407	Special Topics in Open Source Software	3(3+0)
CSEC-8408	Advanced Research Methods	3(3+0)
CSEC-8409	Evolutionary Computation	3(3+0)
CSEC-8410	Special Topics in Machine Learning	3(3+0)
CSEC-8411	Special Topics in Artificial Intelligence	3(3+0)
CSEC-8412	Advanced Optimization Methods	3(3+0)
CSEC-8413	Special Topics in Data Mining	3(3+0)
CSEC-8414	Peer-To-Peer Systems	3(3+0)
CSEC-8415	Ubiquitous Information Interaction	3(3+0)
CSEC-8416	Evolution and Re-Engineering	3(3+0)
CSEC-8417	Program Comprehension and Reverse Engineering	3(3+0)
CSEC-8418	Software Refactoring	3(3+0)
CSEC-8419	Advance Software Architecture	3(3+0)
CSEC-8420	Category Theory	3(3+0)
CSEC-8421	Concrete Mathematics	3(3+0)
CSEC-8422	Number Theory and Cryptography	3(3+0)
CSEC-8423	Harmonic Analysis	3(3+0)
CSEC-8424	Randomized Algorithms	3(3+0)
CSEC-8425	Spectral Graph Theory	3(3+0)
CSEC-8426	Wavelets	3(3+0)
CSEC-8427	Information Theory I	3(3+0)
CSEC-8428	Special Topics in Big Data Analysis	3(3+0)
CSEC-8429	Advance topics in Fog Computing	3(3+0)
CSEC-8430	Advance topics in Cloud Computing	3(3+0)
CSEC-8431	Natural Language Engineering	3(3+0)
CSEC-8432	Advance Natural Language Engineering	3(3+0)
CSEC-8433	Fundamentals of Text Processing	3(3+0)
CSEC-8434	Fundamentals of Speech Recognition	3(3+0)

BS
COMPUTER SCIENCE

CMPC-5201 **Programming Fundamentals** **4(3+1)**

In this course the student will gain a broad understanding of modern computer programming. Students will acquire introductory skills in problem analysis, solution design, and program construction. Through practical programming activities, the student will gain an appreciation of the nature and history of computer programming. Student will also gain hands on practical skill of problem solving by using computer programming languages. Upon successful completion of this course, students will able to understand how data are represented, manipulated, and stored in a computer. In this course we will Categorize different programming languages and their uses. Use, implement, and evaluate fundamental data structures and associated algorithms; create, implement, debug, and evaluate algorithms for solving substantial problems, including recursive, using divide-and-conquer and via decomposition; select and implement an abstract data type for a given problem. Students will also Develop, understand, test, and evolve substantial programs using a modern IDE, and associated configuration tools.

Contents

1. Overview of Computer Programming
2. Principles of Structured and Modular Programming: Algorithms, Pseudo code, flowcharts.
3. Unary and Binary (arithmetic, relational, arithmetic assignment) operators.
4. Decision Statements: if statement, if-else statement, Multi if-else-if statement. Nested if-else statements, Switch Statement.
5. Decision Statements: Conditional operator, Logical Operators. Program Control.
6. Loops: while loop, do while loop.
7. Nested loop structures, Control Statements, break and continue Statements, Logical Operators.
8. Arrays, Searching techniques, Sorting Arrays: selection sort, bubble sort.
9. Strings: String Library Functions Characters and Strings: Fundamentals of Strings and Characters.
10. Function declaration, definition, Passing Arguments to functions, Returning values.
11. Functions: Passing arrays and strings to functions. Inline functions, Default arguments, Local and global variables.
12. Pointers and their purpose. Pointer expressions. Pointers and arrays, Pointers in functions.

Recommended Texts

1. Deitel, P. J., & Deitel, H. M. (2008). *C++ how to program*. New York: Pearson Prentice Hall.
2. Kochan, S. G. (2011). *Programming in objective-C*. Boston: Addison-Wesley Professional.

Suggested Readings

1. Nickols, F., & Lin, Y. J. (2019). *Precision programming of roving robots*. New York: ASME.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CMPC-5202 **Object Oriented Programming** **4(3+1)**

This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Identify the objects & their relationships to build object-oriented solution. This course also relates C++ to GUI, databases, and real-time programming. Student can also perform lab to enhance their skills in object-oriented programming and they can implement the advance programming techniques to solve the complex and real time problems. Introduction to object-oriented programming. Emphasis on the fundamentals of structured design with classes, including development, testing, implementation, and documentation. Includes object-oriented programming techniques, classes, and objects The Java programming language is used as the teaching vehicle for this course. After the successful completion of course, the students will be able to Acquire knowledge of underlying concepts of object-oriented paradigm. They will also learn Develop an understanding of object-oriented design artifacts and their mapping to object-oriented programming and able to Design and implement object-oriented solutions for small systems involving single/multiple objects.

Contents

1. Introduction to Classes, Objects and Strings, Functions, Initializing Objects with Constructors,
2. Class Templates array and vector; Catching Exceptions, Arrays, Range-Based for Statement, Sorting and Searching arrays, Multidimensional arrays.
3. Classes - Throwing Exceptions: Class Scope and Accessing Class Members, Access Functions and Utility Functions, Constructors, Destructors, Default Member wise Assignment, const Objects and const Member Functions, Composition. Operator Overloading, Overloading Binary Operators,
4. Inheritance, Base Classes and Derived Classes, Constructors and Destructors in Derived Classes, public, protected and private Inheritance, Software Engineering with Inheritance.
5. Polymorphism, Type Fields and switch Statements, Abstract Classes and Pure virtual Functions, Virtual Functions and Dynamic Binding.
6. Stream Input/Output, Streams, Stream Output, Input, Unformatted I/O Using read, write and outstream Manipulators, Stream Format States and Stream Manipulators, Stream Error States, Tying an Output Stream to an Input Prentice Hall File Processing: Files and Streams, Creating a Sequential File, Reading and writing Data to a Sequential File, Updating Sequential Files, Random-Access Files, Creating a Random-Access File, Reading and Writing Data Randomly to a Random-Access File.
7. Standard Library Containers and Iterators: Introduction to Containers, Iterators, Algorithms, Sequence Containers, Associative Containers, Container Adapters, Class bit set.
8. Exception Handling, Stack Unwinding

Recommended Texts

1. Deitel & Deitel (2017). C++ *how to program*, (10th ed.). New York: Pearson Prentice Hall.
2. Lafore, R. (1997). *Object-oriented programming in C++*. New York: Pearson Education.

Suggested Readings

1. Paul Deitel (2014). *Java: how to program*, (9th ed.). New Jersey: Pearson Prentice Hall.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CMPC-5101 **Discrete Structures** **3(3+0)**

The course provides a solid theoretical foundation of discrete structures as they apply to computer science problems and structures. The students will learn how to use mathematical notation and solve problems using mathematical tools. In this course student can also learn the Advance counting techniques and its applications from real world. The students will also learn in this course about advance about the different advanced counting techniques and representation of graphs and trees. Students completing this course will be able to express a logic sentence in terms of predicates, quantifiers, and logical connectives. Students completing this course will be able to apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction. The principal topics presented in this course are logic and proof, induction and recursion. Students will develop the mathematical foundations necessary for more specialized subjects in computer science.

Contents

1. Logic: Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Methods of Proof.
2. Sets & Functions, Sequences and Summations.
3. Algorithms: The Growth of Functions, Complexity of Algorithms, the Integers and Division, Matrices.
4. Number Theory and Cryptography.
5. Advanced Counting Techniques: Recurrence Relations, Solving Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion & its Application.
6. Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.
7. Graph: Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.
8. Trees: Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.
9. Sorting: Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort.
10. Hashing: Hash Functions, Collision Resolution, Deletion.
11. Memory Management: Garbage Collection

Recommended Texts

1. Rosen, K. H., & Krithivasan, K. (2012). *Discrete mathematics and its applications: with combinatorics and graph theory*. San Francisco: McGraw-Hill Education.
2. Johnsonbaugh, R. (1986). *Discrete Mathematics revised edition*. (8^{th} ed.). New York: Pearson Prentice Hall.

Suggested Readings

1. S. B. Maurer and A. Ralston A K Peters. (2008). *Discrete algorithmic mathematics*. (3^{rd} ed.). Boca Raton: CRC Press.

CMPC-5203 **Data Structures and Algorithms** **4(3+1)**

This course provides the basic knowledge of data structures & algorithms and understand the concepts of various data structures and use them in different applications. Students will be able to solve, analyze and evaluate the problems using different data structures and algorithms. Demonstrate& apply independently the various forms of data structures and algorithms. Students will able to formulate new solutions for programing problems or improve existing code using learned algorithms and data structures. Evaluate algorithms and data structures in terms of time and memory complexity of basic operations. The topics covered will be similar to those found in introductory algorithms and data structures courses in computer science departments across the world: sorting and searching algorithms, categorizing efficiency in time and memory use, linked list and tree data structures, hash tables, stacks and queues. You will be able to use and design linked data structures with the analysis skills to measure performance of the algorithms.

Contents

1. Introduction to Basic Object Orientation concepts and to Algorithm's Performance Analysis and Measurement (Big Oh Notation)
2. ADT, Basic Operations, Reading, Writing, Insertion, Deletion, Merging, Binary Search.
3. Introduction to Sorting types and Techniques, Logical and Algorithmic Implementation of Bubble, Selection Sort, Insertion, Quick Sort, Merge Sort.
4. The Stack ADT, Expressions, Postfix Notation, Infix to postfix, postfix evaluation.
5. Introduction to Recursion, Examples of Recursion, Writing Recursive Programs
6. The Queue ADT and Its Applications, Variation of Queue ADT i.e. Circular Que.
7. Priority Queues, Introduction to Pointers, Linear single Link
8. Linked Stacks and Queues, Linear Doubly Linked list
9. Circular Lists: Implementation of queues and stacks, Doubly Link List
10. Introduction to Trees, Introduction to Binary Tree ADT, Mathematical properties, Linked Implementation of Binary Trees
11. Binary Search Tree, Implementation and Applications of BSTs
12. Heaps as Priority Queues, Introduction to Balanced and AVL Trees, Heap Sort.
13. Hashing, Overflow Handling, Open Addressing, Chaining
14. Introduction to graph, Adjacency Matrix representation of graph and Adjacency list
15. Elementary Graph Operations, DFS, BFS, Spanning Trees
16. Shortest path algorithms: Dijkstra Algorithm, Minimum Cost Spanning Trees.

Recommended Texts

1. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). *Data structures and algorithms in C++*. Hoboken: John Wiley & Sons.
2. Drozdek, A. (2012). *Data structures and algorithms in C++*. Boston: Cengage Learning.

Suggested Readings

1. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms,* (3rd ed.). Cambridge: MIT Press.

CMPC-6102 **Information Security** **3(3+0)**

The course introduces students to information and communication technologies and their application in the workplace. How vulnerable networks are and the precautions to prevent the possible attacks on Network. The course introduces students to information and communication technologies and their application in the workplace. Objectives include basic understanding of computer software, hardware, and associated technologies. How computers can be used in the workplace, how communications systems can help boost productivity, and how the Internet technologies can influence the workplace and taking into account security perspectives of networks. In this course students learn basics of information security, in both management aspect and technical aspect. Students understand of various types of security incidents and attacks, and learn methods to prevent, detect and react incidents and attacks. Students will also learn basics of application of cryptography which are one of the key technologies to implement security functions. After completing the course, students will have gained an awareness of key information security principles regarding information, confidentiality, integrity and availability.

Contents

1. Information security foundations, security design principles; security mechanisms.
2. Vulnerabilities and protections: Malwares.
3. Hash functions, Digital signatures, Key management.
4. Authentication: Access control, Symmetric cryptography: Symmetric Encryption, DES,
5. AES, Cipher Block Modes.
6. Asymmetric cryptography: HMAC, The RSA, Diffie- Hellman and Other Algorithms Software security
7. Database security: The need, DBMS Relational Databases, SQL Injection Attacks,
8. Network security: Secure E-Mail and S/MIME, Domain Keys Identified Mail, Secure Sockets Layer (SSL)
9. Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, Wireless Security, Mobile Device Security
10. Firewalls
11. Intrusion detection: Intruders, Intrusion Detection, Analysis Approaches, Types of IDS.
12. Security policies, policy formation and enforcement, risk assessment.
13. Cybercrime, law and ethics in information security
14. Privacy and anonymity of data

Recommended Texts

1. Vermaat, M. E., Sebok, S. L., Freund, S. M., Campbell, J. T., & Frydenberg, M. (2017). *Discovering Computers© 2018: Digital technology, data, and devices*. Toronto: Nelson Education.

Suggested Readings

1. Bhattarai, R., Joyce, G., & Dutta, S. (2016, July). *Information security application design*. Cham: Springer.

CMPC-6201 **Operating Systems** **4(3-1)**

This course includes the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems. Analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions. Demonstrate the knowledge in applying system software and tools available in modern operating systems. By the end of the course you should be able to describe the general architecture of computers describe, contrast and compare differing structures for operating systems understand and analyze theory and implementation of processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files. In addition, during the practical exercise and associated self-study, students will become familiar (if not already) with the Java Language Students will be able to explain the role of the operating system as a high-level interface to the hardware and use OS as a resource manager that supports multiprogramming.

Contents

1. Introduction: Over view of: Operating Systems, Operating-System Structure, Operating-System Operations, Process management, Memory Management, Storage Management, Protection and Security, Protection and Security, Distributed Systems, Special-Purpose Systems, Computing Environments.
2. Operating-System Structures: Operating-System Services, Operating-System Structure, User Operating-System Interface, Virtual Machines, System Calls, Operating-System Generation, Types of System Calls, System Boot, System Programs.
3. Processes: Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, Communication in Client- Server Systems. Threads
4. CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Thread Scheduling, Algorithm Evaluation.
5. Process Synchronization: Background, Monitors, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization.
6. Deadlocks
7. Main Memory: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium.

Recommended Texts

1. Abraham Silberschatz (2018). *Operating systems concepts*, (10th ed.). Hoboken: Wiley.
2. Andrew S. Tanenbaum (2014). *Modern operating systems*, (4th ed.). New York: Pearson Prentice Hall.

Suggested Readings

1. William Stallings (2017). *Operating systems, internals and design principles*, (9th ed.) New York: Pearson Prentice Hall.
2. Robbins, S.P. & Coulter, Mary. (2008). *Management,* (10th ed.). Mary: Prentice Hall.

CMPC-5204 **Database Systems** **4(3-1)**

This course includes fundamental database concepts. Design conceptual, logical and physical database schemas using different data models. Identify functional dependencies and resolve database anomalies by normalizing database tables. Use Structured Query Language (SQL) for database definition and manipulation in any DBMS. Students can also learn about the role of Database administrator in the industry how he will perform various operations on the different large and medium size databases. the course provides a solid technical overview of database management systems, using a current database product as a case study. In addition to technical concerns, more general issues are emphasized. These include data independence, integrity, security, recovery, performance, database design principles, and database administration. Familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing. Upon successful completion of this course, students should be able to explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.

Contents

1. Databases Overview: Basic Concepts and Definitions, Traditional File Processing & Database Approach
2. Data Models, Advantages of Database Approach, Costs and Risks of Database Approach.
3. E-R Model, Modelling Rules of the Organization, Modelling Entities and Attributes, Modelling Relationships
4. Enhanced E-R Model, Representing Supertypes and Subtypes
5. Relational Data Model, Integrity Constraints, Transforming EER Diagrams into Relations
6. Introduction to Normalization, First Normal Form, Second Normal Form, Third Normal Form, Functional Dependencies and Keys
7. The Physical Database Design Process, Designing Fields
8. De normalizing and Partitioning Data, File Organizations (Heap, Sequential, Indexed, Hashed).
9. Introduction to SQL, The SQL Environment, Defining a Database in SQL
10. Basic SQL Commands for Processing Single Tables
11. SQL Commands for Processing Multiple Tables using Joins and Subqueries
12. Client/Server Architectures, Databases in a Two-Tier Architecture, Three-Tier Architectures
13. Web Application Components, Databases in Three-Tier Applications
14. The Roles of Data and Database Administrators, Database Backup and Recovery.
15. Controlling Concurrent Access, Serializability, Locking Mechanisms, Data Dictionaries and Repositories

Recommended Texts

1. Hoffer, J. A., Ramesh, V., & Topi, H. (2011). *Modern database management*, (12th ed.). New York: Prentice Hall.
2. Elmasri, R., & Navathe, S. B. (2015*). Fundamentals of Database Systems* (7th ed.). London: Pearson Education.

Suggested Readings

1. Silberschatz, A. Korth, H. F., & Sudarshan, S. (1997). *Database system concepts* (Vol. 5). New York: McGraw-Hill.

CMPC-5103 **Software Engineering** **3(3+0)**

This course covers the fundamentals concepts of project management, important requirements modeling, fact-finding techniques. At the end of course students will have knowledge of software configuration management. Describe software engineering layered technology and Process frame work. A general understanding of software process models such as the waterfall and evolutionary models. Understanding of software requirements and the SRS documents. Understanding of the role of project management including planning, scheduling, risk management, etc. Describe data models, object models, context models and behavioral. This course will clear general understanding of software process models such as the waterfall and evolutionary models. An introduction to object-oriented software development process and design. Topics include: iterative development, interpretation of requirements and use case documents into code; application of design notation in UML and use of commonly-used design patterns. Student will also have chance to learn Current industry-strength programming languages, technologies and systems feature highly in the practical components.

Contents

1. The Nature of Software, Web Apps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths.
2. Process Models
3. Component Based Development, The Formal Methods Models, Agile Development.
4. Introduction to Systems Analysis and Design, Business Information Systems. Information System Components, Types of Information Systems, Evaluating Software.
6. Requirements Modelling Strategies, Difference between Structured Analysis and Object Oriented Analysis; Difference between FDD Diagrams & UML Diagrams
7. Data & Process Modelling, Diagrams
8. Design Within the Context of Software Engineering, The Design Process, Design Concepts, Design Models: Data Design Elements.
9. Architecture Design Elements, Interface Design Elements
10. System Architecture, Architectural Styles, User Interface Design
11. Software Quality Assurance Software Testing Strategies
12. Validation Testing, System Testing, Internal and External View of Testing
13. Introduction to Project Management, Gantt Chart, Risk Management, Software Risks
14. Maintenance and Reengineering.

Recommended Texts

1. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. London: Palgrave macmillan.
3. Sommerville, I. (2016). *Software engineering GE*. NJ: Pearson Prentice Hall.
4. Shelly, G. B., & Rosenblatt, H. J. (2011). *Systems analysis and design*. Boston: Cengage Learning.

Suggested Readings

1. Shelly, G. B., & Rosenblatt, H. J. (2011). *Systems analysis and design.* Boston: Cengage Learning.

CMPC-6202 Computer Networks 4(3+1)

This course includes the key terminologies and technologies of computer networks, Explain the services and functions provided by each layer in the Internet protocol stack. Identify various internetworking devices and protocols, and their functions in a network. Analyze working and performance of key technologies, algorithms and protocols. Build Computer Network on various Topologies. The course will prepare students to plan and implement a network. Also includes peer-to-peer networks, the client-server model, network operating systems, and an introduction to wide-area networks. The network and implementation tools may vary to meet current development trends. Upon completing the course, the student will be familiar with the basics of data communication, with various types of computer networks. They will also gain have experience in designing communication protocols. They can Understand the principles and concepts on computer networks, understand general-purpose computer networks, Master the computer network applications and Master the knowledge on designing and building a complete system.

Contents

1. Introduction to networks and protocols architecture.
2. Basic concepts of networking, network topologies and the Internet.
3. Layered architecture and the OSI model.
4. Physical layer functionality, data link layer functionality and the TCP/IP protocol architecture.
5. Multiple access techniques, WAN Technologies and protocols, circuit switching and packet switching.
6. Wireless networks, Cellular Network Generations and LTE-Advanced.
7. LAN technologies, LAN protocol architecture and virtual LANs.
8. MAC addressing.
9. Networking devices, bridges, hubs and switches.
10. Network layer protocols, Principles of Internetworking, IPv4 and IPv6.
11. IP addressing, Internet Protocol Operation, virtual private networks and IP Security
12. Transport layer protocols, ports and sockets and connection-oriented transport protocol mechanisms.
13. Routing protocols OSPF, EIGRP, RIP and routing in packet-switching networks.
14. Connection establishment, flow and congestion control, effects of congestion, TCP congestion control and datagram congestion control protocol.
15. Application layer protocols, electronic mail (SMTP and MIME), Web Access: HTTP and DNS,
16. Latest trends in computer networks, real-time traffic and voice over IP.

Recommended Texts

1. William Stallings, *Data and computer communications,* (10th ed.) New York: Pearson Prentice Hall.
2. Kurose, J. F., & Ross, K. W. *Computer networking: a top-down approach.* Boston: Addison-Wesley Professional.

Suggested Readings

1. Coombs, C. A., Forouzan, B. A., & Fegan, S. C. (2001). *Data Communications and networking*. San Francisco: McGraw-Hill.

CMPC-6701 **Capstone I** **3(0+3)**

The Capstone Project is an opportunity for participants to put their leadership competencies into practice. Capstone Project allows the student to identify and develop a project that puts into practice the leadership skills and competencies learned during the courses of study. In Capstone Project I students are working in a group form and finalize the Project Proposal. On approval of Project proposal from Project Coordinator group has to work on finalizing the Software Requirement Specifications. In this phase groups are utilizing the Software Engineering guidelines to finalize the Software Requirement Specifications. On finalization of Requirement documentation software Design process is initiated. In this guideline of software design specifications are followed. By developing Project potters and videos groups can improve and sharpen their multimedia skills. Posters and Videos competitions are held regularly to improve competition among project groups. At the end each group has to give presentation and explain in detail all the work completed during semester.

Contents

1. Identifying Project Area
2. Filtering and Finalizing Project
3. Project Proposal
4. Identifying Project Scope
5. Finalizing project Description
6. Non-Functional Requirements
7. Functional Requirements
8. Identifying Project High level Plans
9. Overall project Description
10. System Architecture
11. Literature Review (In case of Research base project)
12. Software Cost, Time and Effort Estimations
13. Identifying main use cases.
14. Finalizing Software Development tool
15. Project Poster Development
16. Project Video Development

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction.* Switzerland: Springer International Publishing.
2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addison-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc Graw Hill
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York: Pearson.

CMPC-6702 **Capstone II** **3(0+3)**

Capstone Project II is a continuation of Capstone Project I. This course provides students with the opportunity to apply the knowledge and skills acquired in their courses to a specific problem or issue. To allow students to extend their academic experience into areas of personal interest, working with new ideas, issues, organizations, and individuals. Main focus of this course is development of Capstone Project I and according to details and area finalized. Group are developing functional requirements and collaborate with the project supervisors and coordinators throughout the semester. At the end of project groups has to present running project to the Project evaluation Committee. In Capstone Project II student can consult with the supervisor or expert from industry. Capstone Project II engages the students with real world market-oriented problems and complexities. Groups has to put in extra effort along with final semester courses to cover the complexities and deadlines of Capstone Project II.

Contents

1. System Sequence Diagram,
2. Domain Model
3. State Chart Diagrams and Implementation
4. Modeling Generalization, Design Class Diagram, Mapping Model to Domain Model
5. Implementation of Design Class Diagram, Coding patterns, Mapping Design to Code
6. Implementation according to Project Area, Web base application, Android/IOS Development, Research base project, Network base project development, Cloud based project, Hardware supported, IOT based, AI and Machine Learning based projects.
7. Implementation design according to project development area.
8. For Database oriented projects details in points 9-10 are followed.
9. SQL Commands for Processing Multiple Tables using Joins and Sub queries.
10. Client/Server Architectures, Databases in a Two-Tier Architecture, Three-Tier Architectures.
11. Web Application Components, Databases in Three-Tier Applications.
12. For Network/cloud based/AI and Machine Learning based projects implementation according to area specification.
13. Project framework development according to specific area.
14. Applying Software Quality Assurance.
15. Software Testing Strategies, Strategic Issues, Test.
16. Validation, System and Integration Testing.

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction.* Switzerland: Springer International Publishing.
2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addison-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc Graw Hill Higher Education.
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York: Pearson.

MATH-5101 Calculus and Analytical Geometry 3(3+0)

This course provides the foundation and basic ground for calculus and analytical geometry background. Understand the foundation and basic ground for calculus and analytical geometry background. To learn fundamentals of mathematics, calculus and analytical geometry. To enable students to apply the ideas to solve problems of practical nature. Have knowledge related to the fundamentals of calculus and analytical geometry. Purpose of this course is to build the student's knowledge of differential/integral calculus of multi-variable functions based on their past experience of differential/integral calculus and analytic geometry of functions of one independent variable. Students will Understand the differentiation integration and their applications. Apply the acquired knowledge to solve problems of practical nature. After the successful completion of course, the students will be able to familiar to real value functions of one and several variables Learn to analyze and solve problems relating analytical geometry, vector analysis & vector calculus and initial value problems.

Contents

1. Real Numbers and the Real Line, Coordinates, Lines, and Increments, Functions, Shifting Graphs, Trigonometric Functions.
2. Limits and Continuity
3. Derivatives: The Derivative of a Function, Differentiation Rules, Rates of Change, Derivatives of Trigonometric Functions, The Chain Rule, Implicit Differentiation and Rational Exponents.
4. Applications of Derivatives: Extreme Values of Functions, The Mean Value Theorem, The First Derivative Test for Local Extreme Values, Graphing.
5. Integration: Indefinite Integrals, Estimating with Finite Sums, Riemann Sums and Definite Integrals, Properties, Area, and the Mean Value Theorem. Substitution in Definite Integrals.
6. Applications of Integrals: Areas between Curves, Finding Volumes by Slicing, Volume of Solids of Revolution. Cylindrical Shells. Lengths of Plan Curves, Areas of Surfaces of Revolution, Moments and Centers of Mass.
7. Transcendental Functions: Inverse Functions and Their Derivatives, Natural Logarithms, The Exponential Function, ax and logax, Growth and Decay, L'Hôpital's Rule, Relative Rates of Growth, Inverse Trigonometric Functions, Derivatives of Inverse Trigonometric Functions
8. Conic Sections, Parameterized Curves, and Polar Coordinates: Conic Sections.
9. Vectors and Analytic Geometry in Space, Vectors in the Plane Dot Products, Vector Valued.

Recommended Texts

1. Calculus Fraleigh, J. B., & Katz, V. (1985). *Calculus with analytic geometry,* (10^{th} ed.). Boston: Addison-Wesley Professional.
2. Calculus Swokowski, E. W. (1979). *Calculus with analytic geometry*. (6^{th} ed.). Pacific Grove CA: Brooks/Cole Publishers.

Suggested Readings

1. Anton, H., Bivens, I. C., & Davis, S. (2009). *Calculus: early transcendentals*. Hoboken, New Jersey: John Wiley.

MATH-5102 **Probability and Statistics** **3(3+0)**

At the end of the course the students will be able to understand the concepts of data analysis, presentation, counting techniques, probability and decision making. It is important because of its direct application in areas such as genetics, finance and telecommunications. It also forms the fundamental basis for many other areas in the mathematical sciences including statistics, modern optimization methods and risk modelling. Students who successfully complete this course should be able to demonstrate understanding of: basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables. How to derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions. They will know how to calculate probabilities, and derive the marginal and conditional distributions of bivariate random variables.

Contents

1. Introduction to Statistics and Data Analysis, Role of Probability, Sampling Procedures; Collection of Data, The Sample Mean and Median, Measures of Variability
2. Discrete and Continuous Data, Statistical Modeling, Scientific Inspection, and Graphical, Types of Statistical Studies, Observational Study, and Retrospective Study.
3. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, the Product Rule, Bayes ‘Rule.
4. Random Variables and Probability Distributions
5. Mean, Variance and Covariance of Random Variables, Chebyshev ‘s Theorem.
6. Binomial and Multinomial Distributions, Hypergeometric Distribution, Negative Binomial and Geometric Distributions, Poisson Distribution and the Poisson Process.
7. Continuous Probability Distributions: Continuous Uniform Distribution, Normal Distribution
8. Fundamental Sampling Distributions and Data Descriptions: Random Sampling.
9. One- and Two-Sample Estimation Problems: Statistical Inference, Classical Methods.
10. Estimating a Proportion, Estimating the Difference between Two Proportions, Single Sample: Estimating the Variance, Estimating the Ratio of Two Variances.
11. One- and Two-Sample Tests of Hypotheses: General Concepts, testing a Statistical Hypothesis, The Use of P-Values for Decision Making in Testing Hypotheses.
12. Single Sample: Tests Concerning a Single Mean, Two Samples: Tests on Two Means, Choice of Sample Size for Testing Means, Graphical Methods for Comparing Means, One Sample
13. One- and Two-Sample Tests Concerning Variances, Goodness-of-Fit Test, Test for Independence and Homogeneity, Multiple Linear Regression and Certain

Recommended Texts

1. Probability Ross, S. M. (2014). *Introduction to probability and statistics for engineers and scientists*. Cambridge, Massachusetts: Academic Press
2. Murray, R., Spiegel, J., & Schiller, R. (2013). *Schaum's outline of probability and statistics*. San Francisco: McGraw Hill Higher Education.

Suggested Readings

1. Haigh, J. (2012). *Probability: A very short introduction* (Vol. 310). Oxford: Oxford University Press.

MATH-5103 **Linear Algebra** **3(3+0)**

This course provides the fundamentals solution for the system of linear equations, operations on system of equations, matrix properties, solutions and study of their properties. Students completing this course will be able to compute the inverse of an invertible matrix. Students completing this course will be able to find the null space of a matrix and represent it as the span of independent vectors. Students completing this course will be able to find the matrix representation of a linear transformation given bases of the relevant vector spaces. Computer software and graphing calculators will be used to enhance the learning and teaching of topics and techniques covered. This course will be creating the abilities in the students to compute the inverse of an invertible matrix. At the end of the course students will be able to find the null space of a matrix and represent it as the span of independent vectors.

Contents

1. Introduction to Vectors: Vectors and Linear Combinations, Lengths and Dot Products, Matrices.
2. Solving Linear Equations: Vectors and Linear Equations, the Idea of Elimination, Elimination Using Matrices, Rules for Matrix Operations, Inverse Matrices.
3. Elimination = Factorization; A = LU, Transposes and Permutations
4. Vector Spaces and Subspaces: Spaces of Vectors, The Null space of A: Solving Ax = 0,
5. The Rank and the Row Reduced Form, the Complete Solution to Ax = B, Independence, Basis and Dimension, Dimensions of the Four Subspaces.
6. Orthogonally
7. Determinants: The Properties of Determinants, Permutations &Cofactors, Cramer's Rule
8. Eigenvalues and Eigenvectors: Introduction to Eigenvalues, Diagonalizing a Matrix, Applications to Differential Equations, Symmetric Matrices, Positive Definite Matrices, Similar Matrices, Singular Value Decomposition (SVD).
9. Applications: Matrices in Engineering, Graphs and Networks, Markov Matrices, Population, and Economics; Linear Programming, Fourier series: Linear Algebra for Functions, Linear Algebra for Statistics and Probability, Computer Graphics. Numerical Linear Algebra: Gaussian Elimination in Practice, Norms and Condition
10. Numbers, Iterative Methods for Linear Algebra.
11. Complex Vectors and Matrices: Complex Numbers, Hermitian and Unitary Matrices, Matrix Factorizations.

Recommended Texts

1. Strang, G., Strang, G., Strang, G., & Strang, G. (1993). *Introduction to linear algebra* (Vol. 3). Wellesley, MA: Wellesley-Cambridge Press.
2. Advanced Zill, D., Wright, W. S., & Cullen, M. R. (2011). *Advanced engineering mathematics*. Sudbury, Massachusetts: Jones & Bartlett Learning.

Suggested Readings

1. Rose, J. S. (1994). *A course on group theory*. North Chelmsford: Courier Corporation.
2. J. B. Fraleigh, (2002). *A first course in abstract algebra*, New York: Pearson Prentice Hall.

PHYS-5101 **Applied Physics** **3 (3+0)**

This course develops the understanding of the fundamental concepts/laws in physics by explaining and discussing the physics as well as their relevance to everyday events and circumstances in a broad interdisciplinary context. It demonstrates teamwork skills/ ability to collaborate by working in groups on a laboratory experiment. At the end of the course the students will be able to understand the basic laws of physics, resistors, inductors, energy sources, Magnetism, electromagnetism circuiting and basics of transistors. The primary objective is to endow the knowledge of a wide variety of electric and magnetic phenomena along with their scientific applications, specifically, in the field of computer science. This course equips the students with the applied concepts of the Physics. Course brushes the basic knowledge of students by starting from the basic concepts and then progresses gradually toward the advance concepts. By the course completion, students would have developed good understanding of Physics fundamentals.

Contents

1. Zero Reference Level, Chassis Ground, Ohm's Law, Formula Variations of Ohm's Law, Graphical Representation of Ohm's Law, Linear Resistor, Non-Linear Resistor, Cells in Series and Parallel.
2. Resistive Circuits.
3. Resistors (5.1-15)
4. Inductors (5.19-21)
5. Capacitors (5.35-48).
6. Energy Sources.
7. Magnetism and electromagnetism
8. Solid State. Atomic structure, Electron distribution of different atoms, Energy bands in solids, Bonds in solids, Conduction in solids, Conductors, Semiconductors and types of semiconductors, Insulators
9. P-N Junction. Formation of depletion layer, Junction or barrier voltage, Forward biased P-N Junction, Forward V/I Characteristics, Reverse biased P-N Junction, Re-verse Saturation Current
10. Optoelectronics Devices. Spectral response of human eye, Light Emitting Diode (LED), Photo emissive Devices
11. DC Power Supplies. Unregulated and Regulated Power Supply, Steady and Pulsating DC Voltages, Rectifiers (17.5-17.8), Filters (17.9-17.2), Voltage Multipliers (17.24-17.30), Silicon Controlled Rectifier SCR (17.33-17.37)
12. The Basic Transistor. Transistor Biasing, Transistor Circuit Configuration.
13. Modulation and Demodulation. Carrier Waves, Modulation, Demodulation or Detection, Comparison between Amplitude Modulation (AM) and Frequency Modulation (FM).

Suggested Readings

1. Rose, J. S. (1994). *A course on group theory*. North Chelmsford: Courier Corporation.
2. J. B. Fraleigh, (2002). *A first course in abstract algebra*, New York: Pearson Prentice Hall.

Recommended Texts

1. Streetman, B. G., & Banerjee, S. (1995). *Solid state electronic devices* (Vol. 4). Englewood Cliffs, NJ: Prentice hall.
2. Malvino, A. P., & Tipler, P. A. (1). *Electronic principles*. Glencoe: McGraw-Hill.

ENGL-5101 **Functional English** **3(3+0)**

The importance of English language cannot be denied. It enriches our thought and culture and provides us with the most important international vehicle of expression. It has opened for us several doors of knowledge for it is the lingua franca of the world and also the language of science, technology, commerce and diplomacy. The main objective of this course is to enhance English language skills of the students and develop their critical thinking. In today ‘s employment market employers are looking for people who can articulate clearly, take and pass on messages, deal with customers effectively, read, understand and follow a wide range of documents and write fluently and accurately, using accepted business conventions of format, spelling, grammar and punctuation. Functional English course is developed to strengthen students ‘these skills which enable them to deal with the practical problems and challenges of life – at home, in education and at work.

Contents

1. Punctuation. Writing Mechanics. Vocabulary
2. Frequently Confused Words, Frequently Misused Words, Phrases.
3. Synonyms, Antonyms, Idioms, General Vocabulary.
4. Use of Articles and One, A Little/ A Few, This, That, Care, Like, Love, Hate, Prefer, Wish, All, Each, Every, Both, Neither, Either.
5. Some, Any, No, None. Interrogatives. INS of Nouns. Prepositions.
6. Possessive, Personal, Reflexive.
7. Relative Pronouns and Clauses.
8. Classes of Verbs. Usage of May, Can, Ought, Should, Must, Have To, Need for Obligation; Must, Have, Will and Should.
9. The Auxiliaries Dare and Used. Present, Past, Future and Perfect Tenses.
10. The Gerund &the Participles. Commands, Requests, Invitations, Advice, Suggestions.
11. The Subjunctive. The Passive Voice; Indirect Speech, Conjunctions, Purpose. Clauses: Noun Clauses.
12. Clauses of Reason, Result, Concession, Comparison, Time. Numerals, Dates, and Weights, Spelling Rules.
13. Phrasal Verbs. Irregular Verbs.

Recommended Texts

1. Thomson, A. J., & Martinet, A. V. (1980). *A practical english grammar*. Oxford: Oxford University Press.
2. Swan, M. (1984). *Basic english usage*. Oxford: Oxford University Press.

Suggested Readings

1. Ross, A., & Greatrex, J. (2001). *A2 english language and literature for AQA B*. Portsmouth: Heinemann.
2. Warriner, J. E. (1988). *English composition and grammar: Introductory course*. San Diego, California: Harcourt Brace Jovanovich.

ENGL-5102 **Communication Skills (English-II)** **3(3+0)**

At the end of the course the students will be able to sensitize their communicative behavior, to enable them to reflect and improve on their communicative behavior/performance to build capacities for self-criticism and facilitate growth to lead students to effective performances in communication. In this course students will learn about paragraph and essay writing to represent their ideas in clear thoughts. This course will demonstrate improved interpersonal skills by identifying and developing a repertoire of strategies for improved communication effectiveness and demonstrate the strategies in oral and written contexts. Students will be able to understand the research methods associated with the study of human communication and apply at least one of those approaches to the analysis and evaluation of human communication. Students will develop knowledge, skills, and judgment around human communication that facilitate their ability to work collaboratively with others. Such skills could include communication competencies such as managing conflict, understanding small group processes, active listening, appropriate self-disclosure, etc.

Contents

1. What Is Communication, The Importance of Communication, The Communication Process, Perspectives in Communication.
2. Internal Representation: Introduction, Internal Representation of Our World.
3. Representational System, Auditory Digital Representational System, Eye Movements as an Indication, Visual Recall, Visual Construct, Auditory Recall, Auditory Construct, Kinesthetic, Internal Auditory, Phrases for Use in Response to Each Representational System.
4. Elements of Communication: Introduction, Face to Face Communication, Tone of Voice, Body Language, Verbal Communication, Physical Communication. Communication.
5. Listening: Real Vs. Introduction, Self-Awareness, Pseudo Listening, Active Listening.
6. Blocks to Listening, Assessing Your Listening, Four Steps to Effective Listening, Total Listing, Becoming an Active Listener.
7. Expressing: The Four Kinds of Expression, Whole Message, Contaminated Messages,
8. Preparing Your Message, Practicing Whole Message, Rules of Effective Expression. Body Language: Body Movement, Spatial Relationships. Paralanguage and Meta-messages:
9. Clarifying Language: Understanding a Model, Challenging the Limits of a Model, Challenging Distortion in a Model. Assertiveness.
10. Effective Written Communication

Recommended Texts

1. Training, M. T. D. (2012). *Effective communication skills*. London: Bookboon.
2. McKay, M., Davis, M., & Fanning, P. (2009). *Messages: The communication skills book*. Portsmouth, New Hampshire: Heinemann.

Suggested Readings

1. Patty Ann, Patty Ann; *Essential communication skills*: (1st ed.). Oxford: Oxford University Press.
2. Robbins, S.P. & Coulter, Mary. (2008). *Management*, (10th ed.). Mary: Prentice Hall.

ENGL-5103 **Technical & Report Writing** **3(3+0)**

Technical writing is a formal, structured and sophisticated writing to fulfill the requirements for a particular field of study. The course aims at providing understanding of writer's goal of writing (i.e. clear, organized and effective content) and to use that understanding and awareness for academic reading and writing. The objectives of the course are to make the students acquire and master the technical writing skills. The course would enable the students to develop argumentative writing techniques. The students would be able to the content logically to add specific details on the topics such as facts, examples and statistical or numerical values. The course will also provide insight to convey the knowledge and ideas in objective and persuasive manner. Furthermore, the course will also enhance the students' understanding of ethical considerations in writing academic assignments and topics including citation, plagiarism, formatting and referencing the sources as well as the technical aspects involved in referencing.

Contents:

1. Discovering and Planning: Discovering Topics; Generating Ideas; Organizing Information; Planning in Digital Environments.
2. Purpose, Thesis, and Audience: Identifying Your Focus and Purpose; Creating a Thesis; Understanding Your Readers.
3. Drafting: Moving from Planning to Drafting, Drafting Collaboratively, Drafting in Digital Environments.
4. Revising, Editing, and Proofreading: Making Major Revisions; Making Minor Revisions; Revising Collaboratively; Revising in Digital Environments; Editing; Editing Collaboratively; Proofreading.
5. Paragraphs: Unfocused Paragraphs; Revising for Focus; Incoherent Paragraphs; Revising for Coherence; Poorly Developed Paragraphs;
6. Clear and Emphatic Sentences: Unclear Sentences; Revising for Clear Sentences; Revising for Variety and Emphasis.
7. Reasoning Critically: Recognizing Critical Reasoning, building a Chain of Reasoning, Representing Your Reasoning.
8. Reading Critically: Reading to Understand; Reading to Respond and Evaluate; Using Journals to Turn Reading into Writing.
9. Speaking Effectively: Oral Presentations, Preparing an Oral Presentation, Managing Speech Anxiety, Fielding Questions.
10. Academic Writing: Social and Natural Sciences: Goals of Writing in the Social and Natural Sciences, Writing Tasks in the Social and Natural Sciences, Types of Writing in the Social and Natural Sciences, Abstract, Informative Report, Lab Report, Research Report.
11. Writing in Online Communities: Online Expectations, E-mail Conventions, Online Communities, Web Pages, Avoiding Plagiarism and Behaving Ethically Online.

Recommended Texts

1. Chris M. Anson, Robert A. Schwegler and Marcia F. Muth *Writer's Companion – The Longman* (2007), London: Pearson Longman.

2. Pickett, N. A., & Laster, A. A. (1996). *Technical english: Writing, reading, and speaking*. New York: Harpercollins College Division.

URCI-5105 **Islamic Studies** **2(2+0)**

Islamic Studies engages in the study of Islam as a textual tradition inscribed in the fundamental sources of Islam; Qur'an and Hadith, history and particular cultural contexts. The area seeks to provide an introduction to and a specialization in Islam through a large variety of expressions (literary, poetic, social, and political) and through a variety of methods (literary criticism, hermeneutics, history, sociology, and anthropology). It offers opportunities to get fully introductory foundational bases of Islam in fields that include Qur'anic studies, Hadith and Seerah of Prophet Muhammad (PBUH), Islamic philosophy, and Islamic law, culture and theology through the textual study of Qur'an and Sunnah.
Islamic Studies is the academic study of Islam and Islamic culture. It majorly comprises of the importance of life and that after death. It is one of the best systems of education, which makes an ethical groomed person with the qualities which he/she should have as a human being. The basic sources of the Islamic Studies are the Holy Qur'an and Sunnah or Hadith of the Holy Prophet Muhammadﷺ. The learning of the Qur'an and Sunnah guides the Muslims to live peacefully.

Contents

1. Study of the Qur'an (Introduction to the Qur'an, Selected verses from *Surah Al-Baqarah, Al-Furqan, Al-Ahzab, Al-Mu'minoon, Al-An'am, Al-Hujurat, Al-Saff*)
2. Study of the Hadith (Introduction to Hadith literature, Selected Ahadith (Text and Translation)
3. Introduction to Qur'anic Studies
4. Basic Concepts of Qur'an
5. History of Quran
6. Basic Concepts of Hadith
7. History of Hadith
8. Kinds of Hadith
9. Uloom –ul-Hadith
10. Sunnah & Hadith
11. Seerat ul-Nabi (PBUH), necessity and importance of Seerat, role of Seerah in the development of personality, Pact of Madinah, Khutbah Hajjat al-Wada' and ethical teachings of Prophet (PBUH).
12. Legal Position of Sunnah
13. Islamic Culture & Civilization
14. Characteristics of Islamic Culture & Civilization
15. Historical Development of Islamic Culture & Civilization
16. Comparative Religions and Contemporary Issues
17. Impact of Islamic civilization

Recommend Texts

1. Hassan, A. (1990). *Principles of Islamic jurisprudence.* New Dehli: Adam Publishers.
2. Zia-ul-Haq, M. (2001). *Introduction to al-Sharia al-Islamia*. Lahore: Aziz Publication.

Suggested Readings

1. Hameedullah, M. (1957). *Introduction to Islam*. Lahore: Sh M Ashraf Publisher.
2. Hameedullah, M. (1980). *Emergence of Islam*. New Dehli: Adam Publishers.

URCP-5106 Pakistan Studies 2(2+0)

The course is designed to acquaint the students of BS Programs with the rationale of the creation of Pakistan. The students would be apprised of the emergence, growth and development of Muslim nationalism in South Asia and the struggle for freedom, which eventually led to the establishment of Pakistan. While highlighting the main objectives of national life, the course explains further the socio-economic, political and cultural aspects of Pakistan's endeavors to develop and progress in the contemporary world. For this purpose, the foreign policy objectives and Pakistan's foreign relations with neighboring and other countries are also included. This curriculum has been developed to help students analyses the socio-political problems of Pakistan while highlighting various phases of its history before and after the partition and to develop a vision in them to become knowledgeable citizens of their homeland. Have an in-depth understanding about the working and impacts of a few major civil engineering projects in Pakistan.

Contents

1. Contextualizing Pakistan Studies
2. Geography of Pakistan
3. Geo-Strategic Importance of Pakistan
4. Freedom Movement (1857-1947)
5. Pakistan Movement (1940-47)
6. Muslim Nationalism in South Asia
7. Two Nations Theory
8. Ideology of Pakistan
9. Initial Problems of Pakistan
10. Political and Constitutional Developments in Pakistan
11. Economy of Pakistan: Problems and Prospects
12. Society and Culture of Pakistan
13. Foreign Policy Objectives of Pakistan
14. Diplomatic Relations
15. Current and Contemporary Issues of Pakistan
16. Human Rights: Issues of Human Rights in Pakistan

Recommended Texts

1. Kazimi, M. R. (2007). *Pakistan studies*. Karachi: Oxford University Press.
2. Sheikh, Javed Ahmad (2004). *Pakistan's political economic and diplomatic dynamics*. Lahore: Kitabistan Paper Products.

Suggested Readings

1. Hayat, Sikandar (2016). *Aspects of Pakistan movement.* Islamabad: National Institute of Historical and Cultural Research.
2. Kazimi, M. R (2009). *A concise history of Pakistan.* Karachi: Oxford University Press.
3. Talbot, Ian (1998). *Pakistan: A modern history*. London: Hurst and Company.

ICTC-5201 Introduction to Information & Communication Technologies 3 (2+1)

The course introduces students to information and communication technologies and their application in the workplace. Objectives include basic understanding of computer software, hardware, and associated technologies. How computers can be used in the workplace, how communications systems can help boost productivity, and how the Internet technologies can influence the workplace. The main objective of the course is to make the students familiar to the basic knowledge of information and communication technologies. Students will be able to independently use information technology tools during the course of the study and, subsequently for searching information and preparing reports, project assignments, presentations, and other materials. The objective is also to present trends in ICT and to prepare students to follow them. They will also learn Open office being used on other operating systems and platforms. Specific software's related to specialization areas are also part of course. Course will also cover Computer Ethics and related Social media norms and cyber laws.

Contents

1. Introduction, Overview and its types.
2. Hardware: Computer Systems & Components, Storage Devices, Number Systems
3. Software: Operating Systems, Programming and Application Software,
4. Introduction to Programming
5. Databases and Information Systems Networks
6. The Hierarchy of Data and Maintaining Data,
7. File Processing Versus Database Management Systems
8. Data Communication and Networks.
9. Physical Transmission Media & Wireless Transmission Media
10. Applications of smart phone and usage
11. The Internet, Browsers and Search Engines.
12. Websites and their types.
13. Email Collaborative Computing and Social Networking
14. E-Commerce
15. IT Security and other issues

Recommended Texts

1. Vermaat, M. E., Sebok, S. L., Freund, S. M., Campbell, J. T., & Frydenberg, M. (2017). *Discovering computers© 2018: digital technology, data, and devices*. Scarborough: Nelson Education.
2. Robbins, S.P. & Coulter, Mary. (2008). *Management,* (10th ed.). Mary: Prentice Hall.

Suggested Readings

1. Timothy J. O'Leary and Linda I. O'Leary (2017) *Computing essentials*. San Francisco: McGraw Hill Higher Education.
2. Fuller, F. (2008). *Computers: understanding technology*. Boulder, Colorado: Paradigm Publishing.

CSSC-6101 **Professional Practices** **3(3+0)**

This course is a graduate-level course of Professional Practices. The course aims to elaborate foundation of Professional Practices as a subject. It focuses on both mainstream and critical approaches to visualize and examining how these topics conceptualize Professional Practices as a field of study. The course explicitly relates Professional Practices as cognate. Students will gain confidence in their ability to communicate by practicing and receiving feedback on business communication skills. Through this critical enquiry process, it is expected that students will further develop their understandings of their own practices and change and develop aspects of these practices. It has a particular emphasis on building professional knowledge of working in a business environment and developing work ready skills in the areas of written and oral communication, intercultural communication, client service, problem solving and self-management. Students will develop strategies to communicate these abilities through career communication, including career documents and interviews. Opportunities for improving academic and workplace language proficiency are embedded in the course.

Contents

1. The Engineering Profession.
2. The Structure of Organizations.
3. Finance and Accounting.
4. Anatomy of a Software House.
5. Computer Contracts.
6. Intellectual Property Rights.
7. The Framework of Employee Relations Law and Changing Management Practices.
8. Human Resource Management and Software Engineering.
9. Health and Safety at Work.
10. Software Liability: Liability and Practice.
11. Computer Misuse and the Criminal Law.
12. Graphical Methods for Comparing Means.
13. Regulation and Control of Personal Information: Data Protection, Defamation and Related Issues.
14. The British Computer Society Code of Conduct.

Recommended Texts

1. Bott, F., Coleman, A., & Rowland, D. (2000). *Professional issues in software engineering*. Boca Raton: CRC Press.
2. Robbins, S.P. & Coulter, Mary. (2008). *Management,* (10th ed.). Mary: Prentice Hall.

Suggested Readings

1. Sara Baase. *A gift of fire: social, legal, and ethical issues for computing and the internet* (3rd ed.). New York: Prentice Hall.
2. Robbins, S.P. & Coulter, Mary. (2008). *Management,* (10th ed.). Mary: Prentice Hall.

CSCC-5201 Digital Logic and Design 4(3+1)

This course Introduces students with digital circuit of large complexity and how such circuits could be built in a methodological way, starting from Boolean logic and applying a set of rigorous techniques. In this course students can also learn about the combinational logic analysis and the working of modern Integrated circuits that are need of modern world. Practical and hands on skills will also help students to face and practice on real time circuits and their working. This course deals with sequential circuits flip-flops, synthesis of sequential circuits, and case studies, including counters, registers, and random-access memories. State machines will then be discussed and illustrated through case studies of more complex systems using programmable logic devices. Different representations including truth table, logic gate, timing diagram, switch representation, and state diagram will be discussed. As a digital design engineer, you may assist in developing cell phones, computers, and related personal electronic devices. In this course, the students will learn the basic concepts of digital circuits and their logical designs.

Contents

1. Number Systems: Decimal Numbers, Binary Numbers, Conversions, Binary Arithmetic, I's and 2's Complements of Binary. Signed Numbers, Arithmetic Operations with Signed Numbers, Hexadecimal Numbers, Octal Numbers, Binary Coded Decimal (BCD), Digital Codes, Error Detection and Correction Codes.
2. Logic Gates. Boolean Algebra and logic Simplification: Boolean Operations and Expressions, Laws and Rules of Boolean Algebra, DeMorgan's Theorem, Boolean Analysis of Logic Circuits, Simplification Using Boolean Algebra, Standard Forms of Boolean
3. Combinational logic Analysis: Basic Combinational Logic Circuits, implementing Combinational Logic, The Universal Property of NAND and NOR Gates, Combinational Logic Using NAND and NOR Gates.
4. Functions of Combinational logic: Basic Adders, Parallel Binary Adders, Ripple Carry versus Look-Ahead Carry Adders, Comparators, Decoders, Encoders. Code
5. Latches, Flip-Flops, and Timers
6. Counters: Asynchronous Counter Operation, Synchronous Counter Operation, Up/Down Synchronous Counters, Design of Synchronous Counters.
7. Shift Registers: Shift Register Functions, Serial In/Serial Out Shift Registers, Serial In/Parallel Out Shift Registers, Parallel In/Serial Out Shift Registers, Parallel In/Parallel Out Shift Registers, Bidirectional Shift Registers, Shift Register Counters.
8. Memory and Storage**,** Programmable Logic: FPGA

Recommended Texts

1. Thomas L. Floyd, Prentice Hall; (2005). *Digital fundamentals,* (9th ed.). New York: Pearson Prentice Hall.
2. Thomas L. Floyd, Prentice Hall (2014). *Digital fundamentals: a systems approach,* (11th ed.). New York: Pearson Prentice Hall.

Suggested Readings

1. M. Morris Mano, Michael D. Ciletti *Prentice* Hall (2018). *Digital Design,* (6th ed.). New York: Pearson Prentice Hall.

CSCC-5202 Computer Organization and Assembly Language 4(3+1)

This course covers the basics of computer organization with emphasis on the lower level abstraction of a computer system. Students will be able to understand digital logic, instruction set. Familiarity with assembly language programming. As an outcome of taking this course, the student will be able to describe the basic architecture of a modern microprocessor, make intelligent comparisons to different types of architectures such as RISC and CISC, stack machines, etc., and show examples of typical machine-level instructions and express numbers in the decimal, binary, and hexadecimal number systems, convert numbers between those systems, and add two numbers and obtain a correct result using two numbers from any one of the systems. Also, the student will be able to describe the implementation of two's complement number representation on typical machines, explain why it was chosen for representing signed numbers and how it relates to the internal arithmetic circuitry, and desk-check binary values by performing two's complement arithmetic with pencil and paper.

Contents

1. Introduction to Assembly Language, 80x86 families; program layout.
2. Data Definitions, Basic Instructions.
3. Unsigned Arithmetic
4. Logic and Bit Operations.
5. Modules
6. Separate Assembly
7. Argument Passing
8. Libraries; Combining Assembly and C Code
9. String Instructions; Arrays Macros; Structures
10. Floating Point Instructions
11. Bit MS-DOS.
12. BIOS Disk Accessing
13. BIOS Keyboard/Video/Graphics
14. Interrupts; TSR Programs.
15. Accessing I/O Ports; 8253 Timer

Recommended Books

1. Irvine, K. R. (2015). *Assembly language for x86 processors*. Boston: Prentice Hall
2. Juola, P. (2006). *Principles of computer organization and assembly language.* Uttar Pradesh: Pearson India

Suggested Reading

1. Triebel, W. A., & Singh, A. (1999). *The 8088 and 8086 Microprocessors: programming, interfacing, software, hardware, and applications: including the 80286, 80386, 80486, and the Pentium*. Boston: Prentice Hall.
2. Hyde, R. (2003). *The art of assembly language.* San Francisco: No Starch Press.

CSCC-6101 **Theory of Automata and Formal Languages** **3 (3+0)**

The course introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expressions, formal language, pushdown automaton, and Turing machine. Not only do they form basic models of computation, they are also the foundation of many branches of computer science, e.g. compilers, software engineering, concurrent systems, etc. Students will learn several formal mathematical models of computation along with their relationships with formal languages. In particular, they will learn regular languages and context free languages which are crucial to understand how compilers and programming languages are built. Also, students will learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms. Throughout this course, students will strengthen their rigorous mathematical reasoning skills. At the end of the course the students will be able to explain the different concepts in automata theory and formal languages. Determine solution to simple automata problems. Apply their understanding of key notions through complex problem solving.

Contents

1. Introduction to Automata: The Methods and the Madness, Introduction to Formal Proof, Inductive Proofs, The Central Concepts of Automata Theory.
2. Finite Automata: Introduction of Finite Automata, Deterministic Finite Automata, Nondeterministic Finite Automata, Finite Automata with Epsilon Tran
3. Regular Expressions and Languages, Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expression.
4. Properties of Regular Languages, Proving Languages Not to Be Regular, Closure Properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata, Context-Free Grammars and Languages.
5. Pushdown Automata: Definition of the Pushdown Automaton, The Languages of a PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata.
6. Properties of Context-Free Languages: Normal Forms for Context-Free Grammars, The Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages, Decision Properties of CFLs, Introduction to Turing Machines.
7. Undecidability: A Language That Is Not Recursively Enumerable, Undecidable Problem That Is RE, Undecidable Problems About Turing Machines, Posts Correspondence Problem, Other Undecidable Problems.
8. Intractable Problems: The Classes P and NP, An NP-Complete Problem, A Restricted Satisfiability Problem.

Recommended Books

1. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). *Introduction to automata theory, languages, and computation*. New York: ACM Sigact News.

Suggested Books

1. Linz, P. (2006). *An introduction to formal languages and automata*. Boston: Jones & Bartlett Learning.
2. Kelley, D. (1995). *Automata and Formal Languages: an introduction*. New York: Prentice-Hall, Inc

CSCC-5101 **Design and Analysis of Algorithms** **3(3+0)**

An algorithm is a set of steps of operations to solve a problem performing calculation, data processing, and automated reasoning tasks. An algorithm is an efficient method that can be expressed within finite amount of time and space. The objective of the course is to teach techniques for effective problem solving in computing. The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problem. In each case emphasis of the algorithm will be used to show the efficiency of the algorithm over the native techniques. At the end of the course the students will be able to design algorithms for problems understanding the core logic of problem-solving Time and Space Complexity of Algorithm. Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Contents

1. Role of Algorithms in Computing, Analyzing Algorithms, Designing Algorithms, Growth of Functions, Asymptotic Notation, Standard Notations and Common Functions.
2. Divide-and-Conquer, Strassen's Algorithm for Matrix Multiplication, Recursion.
3. Recurrences: Substitution Method for Solving Recurrences, Recursion-Tree Method for Solving Recurrences, Master Method for Solving Recurrences.
4. Sorting and Order Statistic, Red-Black Trees: Properties of Red-Black Trees, Rotations, Insertion, Deletion; Minimum Spanning Trees: Introduction, Growing a Minimum Spanning Tree.
5. Dynamic Programming: Elements of Dynamic Programming, Longest Common Subsequence, Optimal Binary Search Trees.
6. Greedy Algorithms: Elements of The Greedy Strategy, Huffman Codes, Matroids and Greedy Methods, Task-Scheduling Problem.
7. Elementary Graph Algorithms, Representations of Graphs, Breadth-First Search, Depth-First Search, Topological Sort.
8. Single-Source Shortest Paths: The Bellman-Ford Algorithm, Single-Source Shortest Paths in Directed Acyclic Graphs, Dijkstra's Algorithm.
9. All-Pairs Shortest Paths: Floyd-Warshall Algorithm, Johnson's Algorithm for Sparse Graphs.
10. Maximum Flow: Flow Networks, Ford-Fulkerson Method, Push-Relabel Algorithms
11. String Matching: Naive String-Matching Algorithm, Rabin-Karp Algorithm, String Matching with Finite Automata, Knuth-Morris-Pratt Algorithm.

Recommended Texts

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms.* Cambridge: MIT press.
2. Levitin, A, (2007). *Introduction to the design & analysis of algorithms.* Boston: Addison-Wesley.

Suggested Readings

1. Sedgewick, R. (1995). *Algorithms in c++.* Boston: Addison-Wesley.
2. Sedgewick, R. (2002). *Algorithms in java.* Boston: Addison-Wesley Professional.

CSCC-6201 **Artificial Intelligence** **3 (3+0)**

This course will introduce the basic principles in artificial intelligence. Students will learn representation schemes, problem solving paradigms, constraint propagation, and search strategies. Explore different areas of AI application such as knowledge representation, natural language processing, expert systems, vision and robotics. At the end of the course the students will be able to: Knowledge of current progresses related to AI. Introduction of many advances' subjects taught at MS and PhD level and applied research. The main purpose of this course is to provide the most fundamental knowledge to the students so that they can understand what the AI is. Due to limited time, we will try to eliminate theoretic proofs and formal notations as far as possible, so that the students can get the full picture of AI easily. Students who become interested in AI may go on to the graduate school for further study. The objectives of this course are to provide graduate comprehensive and in-depth knowledge of AI principles and techniques by introducing AI's fundamental problems, and the state-of-the-art models and algorithms used to undertake these problems.

Contents

1. Introduction: What is AI, Foundations of AI, History of AI. Intelligent Agents: Agents and Environments, The Nature of Environments, The Structure of Agents
2. Problem Solving by Searching: Problem Solving Agents, Searching for Solutions, Uninformed Search Strategies.
3. Breadth-First Search, Depth-First Search, Depth-limited Search, Iterative Deepening, Depth-first Search, Comparison of Uninformed Search Strategies.
4. Informed Search and Exploration: Informed (Heuristic) Search Strategies
5. Constraint Satisfaction Problems: Backtracking Search for CSPs, Local Search for CSPs. Adversarial Search: Games, Minimax Algorithm, Alpha-Beta Pruning.
6. Reasoning and Knowledge Representation: Introductions to Reasoning and Knowledge Representation, Propositional Logic, First Order Logic
7. Inference in First-Order Logic: Inference rules for quantifiers, A first-order inference rule, Unification, Forward Chaining, Backward Chaining, A backward chaining algorithm
8. Introduction to Prolog Programming
9. Reasoning Systems for Categories, Semantic Nets and Description logics, Reasoning
10. Reasoning with Uncertainty & Probabilistic Reasoning: Acting Under Uncertainty
11. Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks.
12. Learning from Observations: Forms of Learning, Inductive Learning, Learning Decision Trees
13. Knowledge in Learning, Explanation-Based Learning, Inductive Logic Programming. Statistical Learning, Neural Networks

Recommended Texts

1. Russell, S., Norvig, P. & Canny, J. (2003). *Artificial intelligence: a modern approach*. New York: Prentice Hall/Pearson Education.
2. Levitin, A, (2007). *Introduction to the design & analysis of algorithms.* Boston: Addison-Wesley.

Suggested Readings

1. Jones, M. (2008). *Artificial intelligence: a systems approach*. Hingham, MA: Infinity Science Press.

2. Lucci, S. & Kopec, D. (2013). *Artificial intelligence in the 21st century: a living introduction*. Dulles: Mercury Learning and Information.

CSCC-6102 Parallel and Distributed Computing 3(3+0)

This course covers a broad range of topics related to parallel and distributed computing, including parallel and distributed architectures and systems, parallel and distributed programming paradigms, parallel algorithms, and scientific and other applications of parallel and distributed computing. In lecture/discussion sections, students examine both classic results as well as recent research in the field. The lab portion of the course includes programming projects using different programming paradigms, and students will have the opportunity to examine one course topic in depth through an open-ended project of their own choosing. Course topics may include: multi-core, SMP, MMP, client-server, clusters, clouds, grids, peer-to-peer systems, GPU computing, scheduling, scalability, resource discovery and allocation, fault tolerance, security, parallel I/0, sockets, threads, message passing, MPI, RPC, distributed shared memory, data parallel languages, Map Reduce, parallel debugging, and applications of parallel and distributed computing. On successful completion of this course students will be able to Develop and apply knowledge of parallel and distributed computing techniques and methodologies.

Contents

1. Overview, Introduction, Parallel and Distributed Computing
2. Parallel and Distributed Architectures, Socket programming
3. Parallel Performance, Shared Memory and Threads
4. Parallel Algorithms
5. Parallel Algorithms, OpenMP
6. Scalable Algorithms, Message Passing
7. MPI and Tera grid
8. Distributed Systems
9. MapReduce
10. Clusters
11. Distributed Coordination, Security
12. Distributed Shared Memory, Peer-to-Peer.
13. Performance analysis and tuning, power,
14. Programming models (data parallel, task parallel, process-centric, shared/distributed memory).
15. Scalability and performance studies, scheduling, storage systems, synchronization,

Recommended Texts

1. Lin, C. (2008). *Principles of parallel programming*. Uttar Pradesh: Pearson Education India.
2. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Saddle River, New Jersey: Prentice-Hall.

Suggested Readings

1. Hwang, K., Dongarra, J. J., & Fox, G. C. (2011). *Distributed and cloud computing: clusters, grids, clouds, and the future internet*. Burlington: Morgan Kaufmann.
2. Levitin, A, (2007). *Introduction to the design & analysis of algorithms.* Boston: Addison-Wesley.

CSCC-6103 **Compiler Construction** **3 (3+0)**

In this course, the students study the principles of compiler design and implementation. The primary emphasis is placed on the organization of a typical compiler pipeline, especially focusing on the stages of a compiler frontend. The course begins with lexical analysis and the construction of scanners, then moves on to various top-down and bottom-up parsing algorithms, semantic analysis and type checking, syntax-directed translation, and intermediate code generation. The course content also includes symbol tables, error recovery, and runtime systems. Furthermore, the course includes an overview of code optimization and target code generation. Each student implements a compiler for a small programming language, usually a subset of a well-known high-level programming language. The students learn to use compiler-compiler tools, including scanner and parser generators. The programming project is split into a few parts to make the development a larger program manageable. At the end of the course the students will be able to understand the basic techniques used in compiler construction. Understand the basic data structures used in compiler construction. Design and implement a compiler using a software engineering approach.

Contents

1. Overview of Compilation: Principles of Compilation, Compiler Structure
2. Scanners: Recognizing Words, Regular Expressions, Implementing Scanners.
3. Parsers: Expressing Syntax, Top-Down Parsing, Bottom-Up Parsing.
4. Context-Sensitive Analysis: Type Systems, Attribute-Grammar Framework
5. Intermediate Representations: Graphical IRs, Linear IRs, Mapping Values to Names.
6. The Procedure Abstraction: Procedure Calls, Name Spaces, Communicating Values Between Procedures, Standardized Linkages.
7. Code Shape: Assigning Storage Locations, Arithmetic Operators, Boolean and Relational Operators, Storing and Accessing Arrays, Character Strings, Structure References, Control-Flow Constructs, Procedure Calls.
8. Code Optimization: Scope of Optimization, Local Optimization, Regional Optimization.
9. Data-Flow Analysis: Iterative Data-Flow Analysis, Static Single-Assignment Form, Inter-procedural Analysis.
10. Scalar Optimizations: Taxonomy for Transformations, Example Optimizations.
11. Instruction Selection: Code Generation, Extending the Simple Tree-Walk Scheme, Instruction Selection via Tree-Pattern Matching and Peephole Optimization.

Recommended Texts

1. Cooper, K., & Torczon, L. (2011). *Engineering a compiler*. New York: Elsevier.
2. Lin, C. (2008). *Principles of parallel programming*. Uttar Pradesh: Pearson Education India.

Suggested Readings

1. Aho, A. V., Sethi, R., & Ullman, J. D. (1986). *Compilers, principles, techniques*. Boston: Addison wesley,
2. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J., & Langendoen, K. (2012). *Modern compiler design*. Berlin/Heidelberg: Springer Science & Business Media.

MATH-5104 **Multivariable Calculus** **3(3+0)**

The goals are to develop the skills to have ground knowledge of multivariate calculus and appreciation for their further computer science courses. In this course student will learn about the Multivariable functions, multiple integral Laplace transformation Fourier analysis and power series. This course enhances the skills of the students in the field of calculus and use of Multivariable in the computing edge courses. This course has big impact on the computer science courses so by learning this course student can sharpen their skills that will help them for advance computer courses like datamining and artificial intelligence. Upon successful completion of this course, students should be able to Understand the basic concepts and know the basic techniques of differential and integral calculus of functions of several variables; They can apply the theory to calculate the gradients, directional derivatives, arc length of curves, area of surfaces, and volume of solid, solve problems involving maxima and minima, line integral and surface integral.

Contents

1. Multivariable Functions and Partial Derivatives: Functions of Several Variables. Limits and Continuity. Partial Derivatives. Differentiability, Linearization, and Differentials. The Chain
2. Multiple Integrals: Double Integrals. Areas, Moments, and Centers of Mass. Double Integrals in Polar Form. Triple Integrals in Rectangular Coordinates. Masses and Moments in Three Dimensions. Triple Integrals in Cylindrical and Spherical Coordinates. Substitutions in Multiple Integrals.
3. Laplace Transforms: Laplace Transform. Inverse Transform. Linearity. First Shifting Theorem (s-Shifting). Transforms of Derivatives and Integrals. ODEs. Unit Step Function (Heaviside Function). Second Shifting Theorem (t-Shifting). Short Impulses.
4. Fourier Analysis: Arbitrary Period. Even and Odd Function. Half-Rang Expansions. Forced Oscillations. Approximation by Trigonometric Polynomials. SturmLiouville Problems. Orthogonal Functions. Orthogonal Series.
5. Power Series, Taylor Series: Sequences, Series, Convergence Tests. Power Series. Functions Given by Power Series. Taylor and Maclaurin Series.
6. Residue Integration: Laurent Series. Singularities and Zeros. Infinity. Residue Integration Method. Residue Integration of Real Integrals.

Recommended Texts

1. Thomas, Wiley (2012). Calculus *& analytic geometry*, (10^{th} ed.). Hoboken, New Jersey: Wiley
2. Erwin Kreyszig (2011). *Advanced engineering mathematics*, (10^{th} ed.). Hoboken, New Jersey: Wiley

Suggested Readings

1. James Stewart (2015). Multivariable *calculus*, (6^{th} ed.). Boston: Cengage Learning publishers.
2. Swokowski, Olinick and Pence (2013). *Calculus and Analytical Geometry*, (6^{th} ed.). Boston: Thomson Learning EMEA, Ltd.

CSSC-5101 **Differential Equations** **3(3+0)**

The course develops student's fundamental skills of solving ordinary differential equations, and developing differential equations for real-world problems. By learning this course student can solve various equations problem including linear equations, nonlinear equations, modeling of different equations and modeling with higher –order differential equations. This course will also enhance the problem-solving skills in the students which will help them in the further computing and programming subjects. Students can adapt mathematical software to solve various problems in partial differential equations. Differential equations course will also strengthen the mathematical understating of the students. This course will focus on the formulation of first and second order partial differential equations (PDEs) for three basic types of hyperbolic, parabolic and elliptic equations. This will equip students with the concepts of partial differential equations and how to solve linear Partial Differential with different methods. Students also will be Introduced to some physical problems in Engineering models that results in partial differential equations.

Contents

1. Introduction to Differential Equations: Definitions and Terminology. Initial-Value Problems. Differential Equations as Mathematical Models.
2. First-Order Differential Equations: Solution Curves without a Solution. Separable Variables. Linear Equations. Exact Equations and Integrating Factors.
3. Modelling with First-Order Differential Equations: Linear Models. Nonlinear Models. Modelling with Systems of First-Order Differential Equations.
4. Higher-Order Differential Equations: Preliminary Theory- Linear Equations. Reduction of Order. Homogeneous Linear Equations with Constant Coefficients. Undetermined Coefficients- Superposition Approach. Undetermined Coefficients- Annihila to Approach. Variation of Parameters. Cauchy-Euler Equation. Solving Systems of Linear
5. Differential Equations by Elimination. Nonlinear Differential Equations.
6. Modeling with Higher-Order Differential Equations: Linear Models: Initial-Value Problems. Linear Models: Boundary-Value Problems. Nonlinear Models.
7. Series Solutions of Linear Equations: Solutions about Ordinary Points
8. Systems of Linear First-Order Differential Equations: Preliminary Theory. Homogeneous Linear Systems. Non-homogeneous Linear Systems. Matrix Exponential.
9. Numerical Solutions of Ordinary Differential Equations: Euler Methods. Runge-Kutta Methods. Multistep Methods.

Recommended Texts

1. Zill, D. G. (2012). *A first course in differential equations with modeling applications*. Boston: Cengage Learning
2. Erwin Kreyszig *Advanced engineering mathematics, (10th ed.).* Hoboken, New Jersey: Wiley

Suggested Readings

1. Ralston, A. & Rabinowitz, P. (2001). *A first course in numerical analysis*. Wellman Ave, North Chelmsford, MA: Courier Corporation.
2. Dahlquist, G., & Björck, Å. (2008). *Numerical methods in scientific computing.* Philadelphia: Society for Industrial and Applied Mathematics

CSSC-6102 **Numerical Computing** **3(3+0)**

Numerical computing is an interconnected combination of computer science and mathematics in which we develop and analyze algorithms for solving important problems in science, engineering, medicine, and business. The goal of this course is to give students an introduction to numeric and algorithmic techniques used for the solution of a broad range of mathematical problems, with an emphasis on computational issues and parallel processing. In addition, students will become familiar with one or more array-oriented numeric programming environments. Student will also learn about the mat lab practical software which will help students in further computing courses. Course will demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems. Apply numerical methods to obtain approximate solutions to mathematical problems. Students can derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.

Contents

1. Introduction and Round off Errors: Floating point systems, Round off error accumulation, The IEEE standard
2. Nonlinear Equations in One Variable: Solving nonlinear equations, Bisection method, Fixed point iteration, Newton's method
3. Review of Linear Algebra: Basic concepts like Vectors and matrix norms, Special classes of matrices, Singular values
4. Direct Methods for solving Linear Systems: Gaussian elimination and backward substitution, LU decomposition, The Cholesky decomposition, Sparse matrices
5. Linear Least Squares Problems: Least squares and the normal equations, orthogonal transformations and QR, Householder transformations.
6. Eigen values and Singular Values: The power method and variants, Singular value decomposition, General methods for computing eigen values and singular values
7. Polynomial Interpolation: Lagrange interpolation, Divided differences and Newton's form, Chebyshev interpolation
8. Fourier Transform: The Fourier transform, Discrete Fourier transform and trigonometric interpolation, Fast Fourier transform
9. Numerical Integration: Basic quadrature algorithms, Composite numerical integration

Recommended Texts

1. Ascher, U. M., & Greif, C. (2011). *A first course on numerical methods*. Philadelphia: Society for Industrial and Applied Mathematics
2. Richard, L., & Burden, J. (2011). *Douglas* fairs, numerical analysis. Boston: Cengage Learning

Suggested Readings

1. Greenbaum, A., & Chartier, T. P. (2012). *Numerical methods: design, analysis, and computer implementation of algorithms*. William Street, Princeton, New Jersey: Princeton University Press.
2. Ralston, A. & Rabinowitz, P. (2001). *A first course in numerical analysis*. Wellman Ave, North Chelmsford, MA: Courier Corporation.

CSSC-5103 Object Oriented Analysis and Design 3(3+0)

Deeper investigation into object-oriented principles, design patterns, and metrics for object-oriented systems. The focus is on developing a maturity in coding and design in order to improve your *software craftsmanship.* Also, an understanding of programming language concepts, an understanding of software engineering principles and the ability to apply them to software design, completion of a large software project, an ability to undertake independent learning, an ability to locate and use technical information from multiple sources. Object oriented Analysis and Design course is developed to strengthen the students in the clear understating of software in diagrams form that will help the software engineer as well as customer to understand software behavior prior to development phase. The course will, furthermore, discuss a variety of perspectives of development of information systems in order explore the relationship between programming, data bases and software engineering. A project will be conducted with the aim to analyze and design an information system in a real-world organization.

Contents

1. Introduction to UML, Unification, UML Diagrams, Unified & Rational Unified Process, RUP Disciplines, Case Study, About Inception, Feasibility and Risk Analysis
2. Understanding Requirements, Requirements Types, Use Case Modeling: Use Case Writing Styles, EBP Guidelines
3. System Use Case Diagram, Use Case Table, Activity Diagram, Supplementary Specifications, Vision Document, Rational Rose Overview
4. Elaboration Phase of RUP; Configuration Management; System Sequence Diagram, Domain Model: Identifying Business Classes, Associations, Attributes
5. Implementation of System Sequence & Domain Model
6. Analysis Use Case Diagram, Implementation of Sequence, Collaboration
7. State Chart Diagrams and Implementation
8. Design Patterns: GRASP: Information Expert, Creator, Cohesion & Coupling
9. Use Case Realization Using GRASP Patterns, Design Model: Determining Visibility
10. Modeling Generalization, Design Class Diagram, Mapping Model to Domain Model
11. Implementation of Design Class Diagram, Coding patterns, Mapping Design to Code
12. More Patterns for Assigning Responsibilities, Polymorphism, Pure Fabrication

Recommended Texts

1. Larman, C. (2012). *Applying UML and patterns: an introduction to object-oriented analysis and design and interactive development*. Chennai: Pearson Education India.
2. Jones, C. (2007). *Estimating software costs: Bringing realism to estimating*. New York: McGraw-Hill Companies.

Suggested Readings

1. Booch, G. (2005). *The unified modeling language user guide*. Chennai: Pearson Education India.
2. Quatrani, T. (2000). *Visual modeling with Rational Rose 2000 and UML*. Boston: Addison-Wesley Professional.

3. Kroll, P., & Kruchten, P. (2003). *The rational unified process made easy: a practitioner's guide to the RUP*. Boston: Addison-Wesley Professional.

CSSC-6104 Computer Architecture 3 (3+0)

This course will provide the students with better understanding of the various levels of studying computer architecture and utilization of computer systems or as foundation for more advanced computer-related studies. This course emphasizes the relationship between technology, hardware and programming systems in the evolution of computer architecture the students will be able to understand structure and behavior of the various functional modules of the computer, understand Instruction Set Architecture design and Micro-architecture, RISC and the CISC type, Pipelining and instruction level Parallelism, explain the I/O sub systems and memory module of computer and Flynn 's Taxonomy. To study the basic organization and architecture of digital computers (CPU, memory, I/O, software). Discussions will include digital logic and microprogramming Memory and RISC machines. Such knowledge leads to better understanding and utilization of digital computers, and can be used in the design and application of computer systems or as foundation for more advanced computer-related studies.

Contents

1. Introduction to a Simple Computer: CPU Basics and Organization, The Bus, Clock, The Input/output Subsystem, Memory Organization and Addressing, Interrupts, MARIE, Instruction Processing, Examples of Computer.
2. Architectures: Intel & MIPs Architecture
3. Instruction Set Architecture: Instruction Format, Instruction Types, Addressing, Instruction Pipelining, ISAs – Intel, MIPS, Java Virtual Machine, CISC and RISC.
4. Micro-architecture: Single-Cycle Processor, Multicycle Processor, Pipeline Processors, DDL Representation, Exceptions.
5. Advanced Micro-architecture: Deep Pipelines, Branch Prediction, Superscalar Processor, Out-of-Order Processor, Register Renaming, Single Instruction Multiple Data, Multithreading and Homogeneous Multiprocessing.
6. Memory: Types of Memory, The Memory Hierarchy, Memory Management
7. Input/output and Storage System: I/O and Performance, Amdahl's Law, I/O architectures, Data Transmission Modes, RAID, Future of Data Storage.

Recommended Texts

1. Harris, D., & Harris, S. (2010). *Digital design and computer architecture*. Massachusetts: Morgan Kaufmann.
2. Null, L., & Lobur, J. (2014). *The essentials of computer organization and architecture.* Burlington: Jones & Bartlett Learning.

Suggested Readings

1. Hennessy, J. L., & Patterson, D. A. (2011). *Computer architecture: a quantitative approach.* Amsterdam: Elsevier.
2. Levitin, A, (2007). *Introduction to the design & analysis of algorithms.* Boston: Addison-Wesley.

CSCC-6105 **Internet Architecture and Protocols** **3(3+0)**

The objective of this course is gaining an understanding of the concepts and techniques used to model and implement communications between processes residing on independent host computers. The course examines the conceptual framework for specifying a computer network - the network architecture, and investigates the set of rules and procedures that mediate the exchange of information between two communicating processes - the network protocols. The Open Systems Interconnection (OSI) Reference Model (OSIRM) provides the conceptual framework for identifying the functions required for communications to take place between processes on autonomous hosts. The OSIRM is presented, and the service definitions and protocol specifications for implementing each of the seven layers of the Reference Model are analyzed in detail. The TCP/IP architecture and protocols that were used for the development of the Internet are presented and compared with OSI. This course will also enhance the network understanding and working of internet architecture.

Contents

1. Introduction. Terms and Concepts. Functioning of the Internet. Review of the TCP/IP
2. Error Recovery Operations: Introduction. Data Transfer across Links. The Basic Operations. Relationship of the Link Layer to the OSI Model.
3. The Link Timers. State Variables and Sequence Numbers. Transmit and Receive Windows. The High-Level Data Link Control (HDLC). The HDLC Frame. The FCS Check. Error. Checking. End-to-End Error Recovery by TCP.
4. Carrier Sense Multiple Access/Collision Detection (CSMA/CD) LANs. CSMA/CD Protocol Stacks. Review of Ethernet. CSMA/CD Frames.
5. Internet Wide Area Networks (WANs): Typical Functions of a WAN Network Layer.
6. IP and ICMP: Attributes of IP. Processing the Datagram. Routing Table. Secondary Addresses in the Table. The IP Header.
7. ICMP. Time-to-Live. Destination Unreachable. Redirect. Router Discovery. Pings.
8. TCP and UDP: Protocol Placement of TCP and UDP.
9. The Point-to-Point Protocol (PPP) and The Layer 2 Tunneling Protocol (L2TP): Introduction to PPP. PPP and Associated Protocols.
10. Routing Protocols: Introduction. The Routing Domain. Multiple Routing Domains.
11. Routing and Forwarding.
12. Protocol Stacks. Gateway. Types of Routing Protocols.
13. Distance Vector. Link State Metric. Design Goals. Static, Stub, and Default Routes.
14. Distance Vector Protocols.
15. Link State Metric Protocols, Shortest Path First Operations.

Recommended Texts

1. Black, U. (2000). *Internet architecture: an introduction to IP protocols*. New York: Prentice Hall PTR.
2. Forouzan, B. A. (2010). *TCP/IP protocol suite*. New York City: McGraw-Hill, Inc.

Suggested Readings

1. Ramamurthy, B., Rouskas, G. N., & Sivalingam, K. M. (Eds.). (2011). *Next-generation internet: architectures and protocols*. Cambridge: Cambridge University Press.

BUSB-5101 Principles of Management 3(3+0)

Formally defined, the principles of management are the activities that "plan, organize, and control the operations of the basic elements of [people], materials, machines, methods, money and markets, providing direction and coordination, and giving leadership to human efforts, so as to achieve the sought objectives of the enterprise. The course will cover topics fundamentals and principles of management, administrative policy, objectives, and procedures and problems of organizational control and leadership. Describe the four management functions of planning, organizing, leading, and controlling. Outline the historical evolution of management theories. Explain how corporate culture and the environments of an organization (general, specific, and global) affect management. Relate the concepts of social responsibility and managerial ethics. Explain how decisions are made within an organization and how those decisions are communicated to the various stakeholders. The course will cover topics fundamentals and principles of management, administrative policy, objectives, and procedures and problems of organizational control and leadership.

Contents

1. Introduction to Managers and Management: What as Management and What Do Managers Do? Defining Management, Management Functions.
2. Management Roles, Management Skills.
3. Organizational Culture and Environment: The Manager: Omnipotent or Symbolic? The Organization ‘s Culture.
4. The Environment - Defining Environment, the Specific Environment.
5. Decision Making the Essence of Manager ‘s Job: The Decision-Making Process, The Rational Decision Maker,
6. Decision Making Styles, Analysing Decision Alternatives – Certainty, Risk.
7. Planning: The Foundations of Planning, The Definition of Planning, Purposes of Planning, Types of Plans.
8. Contingency Factors on Planning, Objectives: The Foundation for Planning.
9. Organization Structure and Design: Defining Organization Structure and Design, Building, the Vertical Dimension of Organizations, Building the Horizontal Dimension of Organizations, the Contingency Approach to Organization Design, Application of Organization Design.
10. Motivation: Motivating Employees, what is Motivation? Contemporary Approaches to Motivation, Contemporary Issues in Motivation, From Theory to Practice.
11. Suggestions for Motivating Employees.

Recommended Texts

1. Robbins, S.P. & Coulter, Mary. (2008). *Management.* (10th ed.). New York: Prentice Hall.
2. Robbins, S.P. & DeCenzo, David A. (2010). *Fundamentals of Management.* (7th ed.). New York: Prentice Hall.

Suggested Readings

1. Charles W. L. Hill and McShane S. (2006). *Principles of management.* (1st ed.). New York City: McGraw-Hill/Irwin.
2. Carpenter M. (2009). *Principles of management.* Boston, MA: Flat World Knowledge.

BUSB-5102 **Human Resource Management** **3(3+0)**

After successfully completing of this course, students will be able to understand the core aspects of Human Resource Management required in 21st Century organizations. Human resources are used to describe both the people who work for a company or organization and the department responsible for managing resources related to employees. The term *human resources* were first coined in the 1960s when the value of labor relations began to garner attention and when notions such as motivation, organizational behavior, and selection assessments began to take shape. The main objective of Human Resource Management is to help the students to acquire and develop skill to design rationale decisions in the discipline of human resource management. An efficient HR manager must guide the work force, influence their behavior and motivate them to conduct maximum towards the achievement of organizational goals. This course focuses on issues and strategies required to select and develop manpower resources.

Contents

1. Managing Human Resources.
2. Understanding the External and Organizational Environments.
3. Ensuring Fair Treatment and Legal Compliance.
4. HR Planning for Alignment and Change.
5. Using Job Analysis and Competency Modeling.
6. Recruiting and Retaining Qualified Employees.
7. Selecting Employees to Fit the Job and the Organization.
8. Training and Developing a Competitive Workforce.
9. Conducting Performance Management.
10. Developing an Approach to Total Compensation.
11. Using Performance-Based Pay to Achieve Strategic Objectives.
12. Providing Benefits and Services for Employees' Well-Being.
13. Risk Management
14. Employee Relations
15. Risk Management, Health, Safety
16. Employee Well-Being.
17. Understanding Unionization
18. Collective Bargaining.

Recommended Texts

1. Jackson, S. E., Luo, Y., & Schuler, R. S. (2003). *Managing human resources in cross-border alliances*. Abingdon: Routledge.
2. Dessler, G., Cole, N. D., & Chhinzer, N. (2015). *Management of human resources: The essentials*. London: Pearson.

Suggested Readings

1. Price, A. (2007). *Human resource management in a business context*. Cheriton: Cengage Learning EMEA.

BUSB-6103 **Organizational Behavior** **3(3+0)**

Study of the behavior of individuals and groups as part of the social and technical system in the workplace. They examine individual and group behavior, communication, conflict and various management styles, motivational techniques and coordination in the work environment and apply these concepts to the development of an organization's human resources. The purpose of one part of this course is to provide a strong conceptual framework for studying, understanding, and applying the theory and practice of organizational behavior. While the other portion explores the nature of organization and organization theory. This course on organization theory and behavior thus provides knowledge that helps people understand, diagnose, and respond to emerging organizational needs and problems of our educational institutions. This course will provide a theoretical foundation that can be used to evaluate organizational phenomena and managerial practices. The purpose of one part of this course is to explores the nature of organization and organization theory while the other portion provides a strong conceptual framework for understanding theory and practice of organization behavior, This course on organization theory and behavior thus provides knowledge that helps people understand, and respond to emerging needs and problems of our educational institutions.

Contents

1. Organizational Behavior: The Quest for People-Centered Organizations and Ethical Conduct
2. Organizational Culture, Socialization, and Mentoring
3. Key Individual Differences and the Road to Success
4. Values, Attitudes, Job Satisfaction, and Counterproductive Work Behaviors
5. Social Perceptions and Attributions
6. Foundations of Motivation
7. Improving Job Performance with Goals, Feedback, Rewards, and Positive Reinforcement
8. Group Dynamics
9. Developing and Leading Effective Teams
10. Individual and Group Decision Making
11. Managing Conflict and Negotiating
12. Communicating in the Digital Age
13. Leadership, Influence, Empowerment, and Politics
14. Organizational Design, Effectiveness, and Innovation

Recommended Texts

1. Miner, J. B. (2005). *Organizational behavior: Essential theories of motivation and leadership.* Armonk, New York: ME Sharpe.
2. Steers, R. M. (1981). *Introduction to organizational behavior*. New York: Goodyear Publishing Company.

Suggested Readings

1. Robbins, S. P., & Judge, T. (2003). *Essentials of organizational behavior*. New York: Prentice Hall.
2. Johnson, C. E. (2019). *Meeting the ethical challenges of leadership: Casting light or shadow*. Thousand Oaks: SAGE Publications.

BUSB-6104 **Business Economics** **3(3+0)**

Business Economics introduces economic concepts and principles which are useful in understanding the economic environment within which businesses. Microeconomics examines how consumers and firms make decisions and how they interact with each other in markets. Macroeconomics looks at the effect of factors such as inflation, exchange rates, interest rates and trade on the well-being and stability of the economy. Economics is not a collection of facts to be written down and memorized. Economics is a way of thinking about the world – and the world is always changing. Economists have developed a set of simple but widely applicable concepts and principles that are useful for understanding economic situations ranging from decisions that individuals make every day to decisions made by firms and governments in complex markets. The objective of this course is to help students learn and understand these concepts and principles and to apply them to a variety of economic situations. This course is mean for understanding core matters of economics so that student shall be able to understand what is going on globally.

Contents

1. Introduction to economics and its history
2. Basic concepts of economic environment
3. Needs, Wants, Demand and Consumer
4. Market and its Environment
5. Introduction to Business and the economic environment
6. The working of competitive markets
7. Demand and the consumer
8. Supply decisions in a perfectly competitive market
9. Pricing and output decisions in imperfectly competitive markets
10. Multinational Environment
11. Business growth and strategy
12. Multinational corporations and business strategy in a global economy
13. The Government
14. The firm
15. The market
16. The economy and business activity
17. National macroeconomic policy
18. The global trading environment

Recommended Texts

1. Sloman, J., Garratt, D., Guest, J., & Jones, E. (2016). *Economics for business*. London: Pearson Education.

Suggested Readings

1. Griffiths, A., & Wall, S. (2008). *Economics for business and management*. London: Pearson Education.

2. McAleese, D. (2004). *Economics for business: Competition, macro-stability, and globalization*. London: Pearson Education.

BUSB-6105 Entrepreneurship 3(3+0)

The course gives students the tools necessary to think creatively, to plan out whether their idea is marketable to investors, guide them through the launch their own business, or to support an employer in launching and growing an entrepreneurial venture. This course provides an understanding of the entrepreneurship process. It exposes them to the concept, practices and tools of the entrepreneurial world. This will be accomplished through a combination of readings, case studies and projects designed to convey the unique environment of the entrepreneurs and new ventures. The course gives students the tools necessary to think creatively, to plan out whether their idea is marketable to investors, guide them on how to launch their own business, or to support an employer in launching and growing an entrepreneurial venture. Entrepreneurship is an interdisciplinary course designed to teach students how to think and act entrepreneurial. Students learn how to start-up and operate a business while in school, thus turning their learning into earning.

Contents

1. Entrepreneurship and the Entrepreneurial Mind-Set.
2. Entrepreneurial Intentions and Corporate Entrepreneurship.
3. Entrepreneurial Strategy: Generating and Exploiting New Entries.
4. Creativity and the Business Idea.
5. Identifying and Analyzing Domestic and International Opportunities.
6. Intellectual Property and Other Legal Issues for the Entrepreneur.
7. The Business Plan: Creating and Starting the Venture.
8. The Marketing Plan.
9. The Organizational Plan.
10. The Financial Plan.
11. Sources of Capital.
12. Informal Risk Capital, Venture Capital, and Going Public
13. Strategies for Growth and Managing the Implication of Growth
14. Succession Planning and Strategies for Harvesting and Ending the Venture.

Recommended Texts

1. Hisrich, R. D. (2014). *Advanced introduction to entrepreneurship*. Edward Elgar Publishing.
2. Dollinger, M. (2008). *Entrepreneurship*. Marsh Publications.

Suggested Readings

1. Kuratko, D. F. (2016). *Entrepreneurship: Theory, process, and practice*. Cengage Learning.
2. Barringer, B. R. (2015). *Entrepreneurship: Successfully launching new ventures*. New Delhi: Pearson Education India.
3. Bygrave, W. D., & Zacharakis, A. (2004). *The portable MBA in entrepreneurship*. New York: John Wiley & Sons.

BUSB-5103 **Financial Accounting** **3 (3-0)**

It is very important to understand the language of the business. A business graduate must understand the language of the business i.e. accounting. Accounting is the system of recording financial transactions with both numbers and text in the form of financial statements. It provides an essential tool for billing customers, keeping track of assets and liabilities (debts), determining profitability, and tracking the flow of cash. The system is largely self-regulated and designed for the users of financial information, who are referred to as stakeholders: business owners, lenders, employees, managers, customers, and others. Stakeholders utilize financial statements to help make business, lending, and investment decisions. The objective is to give student a clear understanding of accounting cycle, nature of assets, liabilities, equities, incomes and expenditure. This will help students in understanding the nature of transactions and their recording cycle for any particular business. The course will serve as a foundation of building advance level understanding at the latter point of study.

Contents

1. Fundamental concepts in accounting
2. Accounting cycle
3. Accounting for Marketable Securities
4. Accounting for Inventories
5. Perpetual and Periodic Inventory systems
6. Cost Flow Assumptions
7. Comprehensive Problem
8. Plant Assets and Depreciation
9. Plant Asset acquisition, Determination of Cost Price
10. Capital and Revenue Classification
11. Methods of Deprecation, Straight Line, Declining Balance, Unit of Out, Sum of Year Digit Depreciation Schedule for Straight Line, Declining Balance Method,
12. Disposal of Fixed Asset, and Accounting for Asset trade in Accounting for Natural and Intangible Assets, Liabilities,
13. Further Readings on Income and Changes in Retained Earnings Partnership Accounting: Setting Up Partners Accounts
14. Distribution of Profits.
15. Admission of a Partner, Dissolution of Partnership when all partners are solvent
16. Statement of Cash Flows Further Readings on Statement of Cash Flows

Recommended Texts

1. Melgs, W. B., & Meigs, R. F. (1988). *Accounting the basis for business decision*. New York: Canadian Catalogguing in Publication Data

Recommended Readings

1. Christine, J. (2017). Principles of financial accounting. (2nd ed.). New York: McGraw-Hill Education
2. Kioko, S., & Marlowe, J. (2016). *Financial strategy for public managers*. (3rd ed.). Montréal, QC: The Rebus Foundation

URCC-5110 Citizenship Education and Community Engagement 3 (1+2)

In recent years, community engagement has become a central dimension of governance as well as policy development and service delivery. However, efforts to directly involve citizens in policy processes have been bedeviled by crude understandings of the issues involved, and by poor selection of techniques for engaging citizens. This course will provide a critical interrogation of the central conceptual issues as well as an examination of how to design a program of effective community engagement. This course begins by asking: Why involve citizens in planning and policymaking? This leads to an examination of the politics of planning, conceptualizations of "community" and, to the tension between local and professional knowledge in policy making. This course will also analyze different types of citizen engagement and examine how to design a program of public participation for policy making. Approaches to evaluating community engagement programs will also be a component of the course. Moreover, in order to secure the future of a society, citizens must train younger generations in civic engagement and participation.

Contents

1. Introduction to Citizenship Education and Community Engagement: Orientation
2. Introduction to Active Citizenship: Overview of the ideas, Concepts, Philosophy and Skills
3. Identity, Culture and Social Harmony: Concepts and Development of Identity
4. Components of Culture and Social Harmony, Cultural & Religious Diversity
5. Multi-cultural society and inter-cultural dialogue: bridging the differences, promoting harmony
6. Significance of diversity and its impact, Importance and domains of inter-cultural harmony
7. Active Citizen: Locally active, globally connected
8. Importance of active citizenship at national and global level
9. Understanding community, Identification of resources (human, natural and others)
10. Human rights, Constitutionalism and citizens' responsibilities: Introduction to human rights
11. Universalism vs relativism, Human rights in constitution of Pakistan
12. Public duties
13. Public responsibilities
14. Social Issues in Pakistan: Introduction to the concept of social problem, Causes and solutions
15. Social Issues in Pakistan (Poverty, Equal and Equitable access of resources, unemployment)
16. Social Issues in Pakistan (Agricultural problems, terrorism & militancy, governance issues)
17. Social action and project: Introduction and planning of social action project
18. Identification of problem, Ethical considerations related to project
19. Assessment of existing resources

Recommended Books

1. Kennedy, J. K. Brunold, A. (2016). *Regional context and citizenship education in asia and europe*. New York: Routledge Falmer.
2. Macionis, J. J. Gerber, M. L. (2010). *Sociology*. New York: Pearson Education.

Suggested Books

1. British, Council. (2017). *Active citizen's social action projects guide.* Scotland: British Council

PSYC-5101 Introduction to Psychology 3 (3+0)

This course has been designed to ensure an effective orientation of students towards the discipline of psychology so that they may come to appreciate the diversity of the subject and its pragmatic significance. This course provides an introduction to the concepts and theories of psychology and to their application to real life situations. Topics include history, research methods, sensation, perception, consciousness, stress and coping, learning, memory, motivation and emotions. Main objectives of the course include to make students familiar with the essential features of human personality; to inculcate a sense of personal relevance of Psychology as a subject with the potential of gaining better insight into one's own self and others. Upon the successful completion of course students will have an introductory knowledge of selected areas of basic psychological enquiry and they will be able to: differentiate between scientific and non-scientific information about human behaviors and mental processes, describe major developments and research methods used in psychology; Explain psychological processes involved in sensation, perception, learning, memory, motivation, emotion, states of consciousness and health; Analyze the variety of factors affecting sensation, perception, consciousness, learning, memory, motivation, emotion, and health; and can apply psychological concepts and principles to situations in everyday life.

Contents

1. Introduction to Psychology: Definition of psychology, Goals of psychology, Major schools of thought in psychology, Major fields of psychology
2. Basic research Methods in Psychology: Survey research, Experimental research
3. Biological Basis of Behavior: Brain and nervous system, Structure and function of major brain areas, Neurotransmitters and their functions
4. Sensation and Perception: Difference between sensation and perception, Principles of perception, Role of perception in human cognition
5. Motivation and Emotion: Concept & Theories of motivation and emotion
6. Learning: Definition of Learning, Types of Learning
7. Memory and Intelligence: Definition and stages of human memory, Types of memory, Concept of intelligence, Basic theories of intelligence
8. Personality development: Concept & Theories; Tips to improve personality
9. Health and Stress, Stress and Coping, Stress, Health, and Coping in the Workplace, Effective Measure to deal with stress and ways to cope.
10. Application of Psychology in Our Social Lives

Recommended Texts

1. Weiten, W. (2017). *Psychology: Themes and variations* (10th ed.). Boston: Cengage Learning.
2. Nolen-Hoeksema, S., & Hilgard, E. R. (2015). *Atkinson and Hilgard's introduction to psychology* (16th ed.). New Dehli: Cengage Learning.

Suggested Readings

1. Flanagan, C., Berry, D., Jarvis, M., & Liddle, R. (2015). *AQA psychology.* London: Illuminate Publishing - Cheltenham.
2. Coon, D., Mitterer, J. O., & Martini, T. S. (2018*). Introduction to psychology: Gateways to mind and behavior* (15th ed.). Boston: Cengage Learning.

CSEC-6101 **Web System and Technology** **3(3+0)**

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). This course will give an overview of Web Systems and Technologies. Students will learn the essential skills of website management; understanding of the basic Internet technology concepts, develop a prototype of interactive World Wide Web applications. This subject will provide students with the principles and practical programming skills of developing Internet and Web applications. It enables students to master the development skill for both client-side and server-side programming, especially for database applications. Students will have opportunity to put into practice the concepts through programming exercises based on various components of client/server web programming. Students will learn the essential skills of website management; understanding of the basic Internet technology concepts, develop a prototype of interactive World Wide Web applications.

Contents

1. Overview of WWW, Web Pages, Web Sites, Web Applications,
2. TCP/IP
3. TCP/IP Application, Services, Web Servers
4. WAMP Configuration.
5. Introduction to HTTP, HTML.
6. HTML5 Tags, Dynamic Web Content, CSS and CSS3
7. Client-Side Programming.
8. JavaScript: Basics, Expressions and Control Flow
9. Functions, Objects, and Arrays, Accessing CSS from JavaScript.
10. Form Handling
11. Server-Side Programing:
12. Programing in PHP
13. Introduction MySQL
14. MySQL Functions.
15. Accessing MySQL via php MyAdmin.
16. Cookies, Sessions, and Authentication.
17. Introduction to XML, Ajax, JQuery.
18. Browsers and the DOM.
19. Designing a Social Networking Site.

Recommended Texts

1. Nixon R, Media O'. (2014). *Learning PHP, MySQL, JavaScript, and CSSC, A Step-by-Step Guide to Creating Dynamic Websites*., Surrey: O'Reilly Media;

Suggested Readings

1. Jeffrey C. Jackson. (2006). *Web Technologies: A Computer Science Perspective*. New York: Prentice Hall.
2. Kumar Roy U. (2011). *Web Technologies*. Oxford: Oxford University Press.

CSEC-6102 **Multimedia System and Design** **3 (3+0)**

The course introduces students with the complete process of multimedia system specifications, formats, design, testing, and prototyping, including the tools and techniques for integrating multimedia content into a product. Understand complete process of multimedia system specifications, formats, design, testing, and prototyping. Learn different tools and techniques for integrating multimedia content into a product. Multimedia offers many career paths that can lead to occupations in such fields as graphic design, web design, animation, audio and video production, and project management. To become competent in any multimedia field, however, you need to learn the fundamental multimedia concepts first. "Multimedia: Making It Work" builds a foundation for success in the discipline of multimedia by introducing you to the multimedia building blocks of text, images, sound, animation, and video while going one step further to develop an understanding of the process of making multimedia. Upon completion of the course the participant will be able to create a well-designed, interactive Web site with respect to current standards and practices.

Contents

1. Introduction
2. Multimedia Authoring and Tools
3. Handling Images
4. Handling Sound
5. Handling Animation
6. Handling Video
7. Making Multimedia
8. Multimedia Skills
9. Planning and Costing
10. Designing and Producing
11. Content and Talent
12. The Internet and Multimedia
13. Designing for the World Wide Web
14. Delivering

Recommended Texts

1. Vaughan, T. (2011). *Multimedia: making it work*. New York: McGraw-Hill.
2. Nian. & Drew, M. (2004). *Fundamentals of multimedia*. New York: Pearson Prentice Hall.

Suggested Readings

1. Chapman, N. & Chapman, J. (2004). *Digital multimedia*. Chichester: Wiley.
2. Austerberry, D. (2005). *The technology of video and audio streaming*. Burlington, MA: Focal Press.
3. Shih, F. (2013). *Multimedia security: watermarking, steganography, and forensics*. Boca Raton, FL: CRC Press.

CSEC-6103 **Big Data Analytics** **3(3+0)**

The main goal of this course is to help students learn, understand, and practice big data analytics and machine learning approaches, which include the study of modern computing big data technologies and scaling up machine learning techniques focusing on industry applications. Mainly the course objectives are: conceptualization and summarization of big data and machine learning, trivial data versus big data, big data computing technologies, machine learning techniques, and scaling up machine learning approaches. This course provides an overview of approaches facilitating data analytics on huge datasets. Different strategies are presented including sampling to make classical analytics tools amenable for big datasets, analytics tools that can be applied in the batch or the speed layer of a lambda architecture, stream analytics, and commercial attempts to make big data manageable in massively distributed or in-memory databases. Learners will be able to realistically assess the application of big data analytics technologies for different usage scenarios and start with their own experiments.

Contents

1. Overview of Big Data: This includes topics such as history of big data, its elements, career related knowledge, advantages, disadvantages and similar topics.
2. Big Data in Real Life: This module focuses on the application perspective of Big Data covering topics such as using big data in marketing, analytics, retail, hospitality, consumer good
3. Technologies for Handling Big Data: Big Data is primarily characterized by distributed data processing frameworks like Hadoop.
4. Hadoop Ecosystem: This includes learning about
5. Hadoop and its ecosystem which includes HDFS
6. Deeply Understanding MapReduce: This module should cover the entire framework of MapReduce and uses of MapReduce.
7. Databases and Data Warehouses: This module should cover all about databases, data warehouses, data mining, and their related introductory knowledge.
8. Data Storage using HDFS: This includes an entire module of HDFS, HBase and their respective ways to store and manage data along with their commands.
9. Big Data Semester Project: Discussion about how to collect real data, store data in HDFS, process distributed data using Apache Spark. Justify how it improves performance.
10. Learn Scala Basics: Learn variables, functions, loops, class, objects, case classes

Recommended Texts

1. VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. " Surrey: O'Reilly Media, Inc.".
2. Stephens-Davidowitz, S., & Pabon, A. (2017). *Everybody lies: Big data, new data, and what the internet can tell us about who we really are*. New York: HarperCollins.

Suggested Readings

1. Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple*. Surrey: O'Reilly Media, Inc.
2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6104 Digital Image Processing 3(3+0)

Give the students a general understanding of the fundamentals of digital image processing. This course will introduce the student to analytical tools which are currently used in digital image processing as applied to image information for human viewing. This course will Develop the students' ability to apply these tools in the laboratory in image restoration, enhancement and compression. They will understand differences between computer vision and image processing and know the basic components of an image processing system with clear Understanding of the basics of the human visual system as they relate to image processing; including spatial frequency resolution and brightness adaption. Specifically, it covers the areas of image acquisition and imaging systems, 2D continuous-time and discrete-time signals and systems, time and frequency representations, sampling and quantization issues, image filtering, convolution and enhancement, image reconstruction and restoration, color image processing, image quality evaluation, image transform and compression, applications and computer implementations.

Contents

1. Overview, Computer imaging systems Introduction to HTTP and Web protocols
2. Image analysis
3. Preprocessing
4. Human visual system
5. Image model
6. Image enhancement
7. Gray scale mods
8. Histogram mod
9. Discrete transforms
10. Fourier discrete cosine
11. Walsh-Hadamard
12. Haar, PCT, filtering
13. filtering, wavelet transform, pseudocolor
14. Image enhancement, sharpening, smoothing
15. Image restoration, overview, system model
16. Noise removal: order filters
17. Image restoration: noise removal: mean & adaptive filters, degradation model, inverse filter
18. Freq. filters, Geometric transforms

Recommended Texts

1. Umbaugh, S. E. (2017). *Digital Image Processing and Analysis: Applications with MATLAB and CVIPtools*. (3rd ed.) Boca Raton: FL CRC Press.
2. Gonzalez, R. C., & Woods, R. (2020). Digital image processing 3rd ed. New York: Prentice Hall

Suggested Readings

1. Burger, W., & Burge, M. J. (2016). *Digital image processing: an algorithmic introduction using Java*. Berlin: Springer.

2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6105 E-Commerce Applications Development 3(3+0)

This course focuses on electronic commerce applications, technologies, and tools which are used to conduct business on the World Wide Web. This course is all about learning basics of developing an E-commerce website. It covers every step of the design and building process involved in creating powerful, extendable e–commerce site. Based around a real–world example involving a web site selling Products, you'll learn how to create and manage a product catalog, build and integrate a shopping cart, and process customer accounts and Stripe/credit card transactions. This course focuses on electronic commerce applications, technologies, and tools which are used to conduct business on the World Wide Web. It reviews foundations of e-commerce, its infrastructure, current business models in business-to-customers (B2C) and business-to-business (B2B) transactions, security and quality assurance. After Completion of the subject student should able to understand the basic concepts and technologies used in the field of management information Systems.

Contents

1. E-Commerce: An overview of e-Commerce, Brick 'N Mortar stores vs Service-based companies, e-Commerce Models, e-Commerce popular sites
2. Planning an e-Commerce Framework: Designing a framework, Patterns, Model View-Controller, Registry, Singleton, Structure, Building a framework, Routing requests.
3. Products and Categories: Product information, Category information, Structuring Content, Versioning, Building products, categories, and content functionality, Routing products and categories.
4. Product Variations and User Uploads: Giving user’s choice, giving users control, shopping.
5. Enhancing the User Experience: The importance of user experience, Search, providing wish lists, Making Recommendations
6. The Shopping Basket: Creating a Basket, Basket Contents, Managing the Basket, Cleaning the Basket.
7. The Checkout and Order Process: The Process, Authentication, Payment Method
8. Shipping and Tax: Shipping Methods, Shipping Costs, Shipping Rules, Tracking, Tax Calculation.
9. Discounts, Vouchers, and Referrals: Discount codes, Purchasable Voucher Codes, Referrals.
10. Checkout: Checkout process consideration, Order process review, Authentication & Confirmation.
11. Taking Payment for Orders: Taking payment, Payment System, Payment gateway, Taking Payment Online, Taking payment offline.
12. User Account Features: User Account Area, Changing Details, Viewing & Managing Orders.

Recommended Texts

1. Peacock, M. (2010). *PHP 5 e-commerce Development*. Mumbai: Packet Publishing Ltd.
2. Laudon, K. C., & Traver, C. G. (2019) *E-commerce: business, technology, society*. (15th ed.). Delhi: Pearson.

Suggested Readings

1. Peacock, M. (2010). *PHP 5 e-commerce Development*. Mumbai: Packet Publishing Ltd.
2. Bidgoli, H. (2002). *Electronic commerce: principles and practice*. Cambridge: Academic press.

SEC-6106 Enterprise Resource Planning Systems 3(3+0)

This course provides students with an understanding of what Enterprise Systems (also commonly-termed as Enterprise Resource Planning Systems, ERPs) are. The course is basically meant for introducing ERP Systems, which provides for integrating planning throughout the business cycle from raw-materials, shop floor control to the related human-resource and finance processes. The course discusses the role of ERP Systems and Software to reduce inventories, waste, scrap, and rework and how to utilize resources efficiently. Functional modules will be given special emphasis to make the users feel the practical world of enterprise planning and its impact on the bottom line of financial statements. This course systematically presents several conceptual and pragmatic methodologies, tools and techniques for various phases of implementation in an enterprise. After learning about what these systems are, we would touch upon why these systems are useful to companies, through which students would get to see the various jobs and positions that are associated with the use and deployment of ERPs.

Contents

1. Introduction to Enterprise Resource Planning Systems.
2. ERP Technology.
3. ERP and Business Process Reengineering.
4. Systems Diagramming and the Process Map.
5. ERP Life Cycle: Planning and Package Selection.
6. ERP Life Cycle: Implementation and Operation and Maintenance.
7. ERP Sales, CRM and Knowledge Management.
8. ERP Financials.
9. Human Capital Management
10. Self-Service and Outsourcing.
11. Manufacturing Systems and Supply Chain.
12. Auditing ERP
13. Business Intelligence
14. Performance Management.

Recommended Texts

1. Bradford, M. (2015). *Modern ERP: select, implement, and use today's advanced business systems*. (3rd ed.). Morrisville: Lulu.
2. Olson, D. L. (2003). *Managerial issues of enterprise resource planning system, (1st ed.).* Pennsylvania: McGraw-Hill, Inc.

Suggested Readings

1. Wagner, B., & Monk, E. (2008). *Enterprise resource planning a complete guide*. Independence Kentucky: Cengage Learning.
2. Srivastava, D., & Batra, A. (2010). *ERP systems*. New Dehli, IK: International Publishing House.

CSEC-6107 **Mobile Application Development** **3(3+0)**

This course will introduce development of the high-quality mobile apps. Android is one of fastest growing mobile application development platform on the markets. Now a day's android is most powerful technology in mobile devices, Objective have technology experts who believe in delivering high performance services to match our clients demand for quality focus and cost effectiveness and turnaround time. Develop mobile applications using current software development environments. Compare the different performance tradeoffs in mobile application development. Students are expected to work on a project that produces a professional-quality mobile application. Projects will be deployed in real-world applications. Course work will include project conception, design, implementation, and pilot testing of mobile phone software applications, using weight loss and physical activity motivation health applications as the target domain. The course aims to elaborate foundation of Mobile Application Development as a subject. It focuses on both mainstream and critical approaches to visualize and examining how these topics conceptualize Mobile Application Development as a field of study.

Contents

1. What is Android? Obtaining the Required Tools, Installing and Configuring the Android SDK Manager, Creating Your First Android Application, Anatomy of an Android Application.
2. The Big Picture, How to Get Started, Your First Android Project, A bit About Eclipse.
3. Understanding Activities, Linking Activities Using Intents, Fragments, Calling Built-In Applications Using Intents, Displaying Notifications.
4. Understanding the Components of a Screen, Adapting to Display Orientation, Managing Changes to Screen Orientation, Utilizing the Action Bar.
5. Using Basic Views, Using Picker Views, Using List View to Display Long Lists.
6. Using Image Views to Display Pictures, Using Menus with Views, Analog Clock and Digital Clock Views.
7. Saving and Loading User Preferences, Persisting Data to Files, Creating and Using Databases.
8. Sharing Data in Android, using a Content Provider, Creating Your Own Content Providers.
9. Sending SMS Messages Programmatically, Getting Feedback after Sending a Message, Sending SMS Messages Using Intent, Receiving SMS Messages, Sending E-mail.
10. Displaying Maps, Getting Location Data, monitoring a Location, Building a Location Tracker.
11. Consuming Web Services Using HTTP, Accessing Web Services Using the Get Method, Consuming JSON Services, Sockets Programming.

Recommended Texts

1. Meier, R. (2012). *Professional android 4 application development,* (3rd ed.). Birmingham: Wrox.
2. Conway, J., & Hillegass, A. (2012). *Ios programming: the big nerd ranch guide*. Atlanta: Big Nerd Ranch Guides.

Suggested Readings

1. Hardy, B., & Phillips, B. (2019). *Android programming: The big nerd ranch guide*. (4th ed.). Atlanta: Big Nerd Ranch Guides.

2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6108 Enterprise Application Development 3 (3+0)

The course is aimed at creating robust enterprise applications using J2EE technologies that allows for rapid change and growth. Develop strong skills to cater Enterprise Application Development needs and challenges. At the end of course students will be able to understanding the structure of modern computer system. Understand the basic principles of implementing computer graphics primitive. Familiarity with key algorithms for modelling and rendering graphic data. This course explores advanced application development techniques in a large enterprise wide setting using Microsoft technologies, ASP.Net, HTML5, CSS3, Bootstrap, and jQuery. Enterprise systems are software applications that automate and integrate all many of the key business processes of an organization. With some understanding of software development, you will learn about current development practices for this type of system and develop relevant skills to apply them to real-world problems. You will develop core skills in object-oriented analysis and design, allowing you to develop software that is fit for purpose, reusable and amenable to change.

Contents

1. Introduction to Enterprise Application Development, Object-Oriented Programming Review, Software Architectures Overview
2. Application and Page Framework, Page Directives, Page Life Cycle, Build Providers Compilation and its techniques, compilation implantations, Application level events.
3. Asp. Net Server Control and Client-Side Scripts, JQuery, JQuery, validation with jQuery.
4. Applying Style to Server Control, HTML Server Control Manipulating Pages and Server Control with JavaScript, Client-Side call-back.
5. Web Server Control
6. Validation Server Control, Client-side validation vs server-side validation
7. Master Pages, Coding Master Pages, Coding Content Pages, Nesting Master Pages.
8. HTML and CSS Design with Asp.Net, Bootstrap and responsive design, Theme and Skins
9. Entity framework and its approaches, LINQ, ADO.Net, Create Database and model and insert/update record using entity framework.
10. Data Binding & Data Management using store procedures, Data Grid view Server Control.
11. SQL Queries, JOINS, built in function, Session and Cookies, User Authentication.
12. Basic of XML, XML Reader and XML Writer.
13. ASP.Net AJAX Control, ASP .Net AJAX Application Control, Server-Side Control, Update Panel Control.
14. Building and Consuming Services, XML Services, Protocol for Web Services

Recommended Books

1. William Penberth (2016), *Beginning asp.net for visual studio 2015*. Birmingham: *Wrox* Press Ltd.
2. Mary Delamater and Anne Boehm. (2016) *Murach's asp.net 4.6 web programming with C# 2015.* Knoll Fresno: CA Mike Murach & Associates.

Suggested Readings

1. Spaanjaars, I. (2014). *Beginning asp .net 4.5. 1: in C# and vb*. Hoboken, New Jersey: John Wiley & Sons.

2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6109 Cloud Computing 3(3+0)

This course gives an introduction to cloud computing and related techniques, issues, ecosystem, and case studies. Students will learn and understand about such fundamental distributed computing "concepts" for cloud computing, how these techniques work inside today's most widely-used cloud computing systems and various research papers will be studied and reviewed to get the idea of current areas of research and study in cloud computing as well as practical work of cloud-based scheduling algorithms. Cloud Computing is a large-scale distributed computing paradigm which has become a driving force for information technology over the past several years. The exponential growth data size in scientific instrumentation/simulation and social media has triggered the wider use of cloud computing services. We will explore solutions and learn design principles for building large network-based systems to support both compute and data intensive computing across geographically distributed infrastructure. Mainstream cloud infrastructure services and related vendor solutions are also covered in detail.

Contents

1. Distributed systems, Characteristics, Design goals, Types of distributed systems
2. Cloud Computing perspectives, Properties and characteristics, Benefits
3. Service and deployment models of Cloud computing,
4. Modern On-Demand Computing, Amazon's Elastic Cloud, Amazon EC2 Service
5. Virtualization, From emulation to virtualization, Goals of virtualization
6. Memory Virtualization: Background, Virtualization Techniques: Emulated TLB, Shadow Page Tables, Hardware supported Memory Virtualization, Nested Page Tables
7. Virtualization Practicum.
8. Cloud Federation: Characterization and Conceptual Model, Voluntary or independent model, Horizontal, Vertical, Hybrid model, Architectural models for cloud federation
9. Presence in the Cloud, Presence Protocols, Leveraging Presence, Presence Enabled
10. Presence Protocols: XMPP, SIMPLE, SIP
11. Privacy and Its Relation to Cloud-Based Information Systems, Privacy Risks and the Cloud, Cloud Security Challenges, Software-as-a-Service Security.
12. End-User Access to Cloud Computing, YouTube, Widgets, YouTube Player APIs, The YouTube Custom Player, YouTube Data API, Zimbra (ZCS), Facebook, Zoho, DimDim Collaborations
13. Mobile Internet Device and the Cloud
14. Cloud, IOT and Fog Computing

Recommended Texts

1. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. Boca Raton: CRC press.
2. Lee, M. L. T. (2007). *Analysis of microarray gene expression data*. Berlin/Heidelberg: Springer Science & Business Media.

Suggested Readings

1. Buyya, R., & Dastjerdi, A. V. (2016). *Internet of things: principles and paradigms*. Amsterdam: Elsevier.
2. Kavis, M. J. (2014). *Architecting the cloud: design decisions for cloud computing service models*.

Hoboken, New Jersey: John Wiley & Sons.

CSEC-6110 **Systems Programming** **3(3+0)**

Introduction to systems programming, system programming languages and application of those languages to systems level problems. Design, write, and test moderately complicated low-level programs using a systems programming language. This course is covered to introduce students the concepts and principles of system programming and to enable them to understand the duties and scope of a system programmer, to provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs, to train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools. The objective of this course is to provide you with a basic understanding of the issues involved in writing system programs on a Linux or Unix system, manipulating system processes, system io, system permissions, files, directories, signals, threads, sockets, terminal, Interrupts, BIOS, Accessing I/O ports etc.

Contents

1. Introduction to the Microsoft Windows Operating System,
2. File Processing, Memory Management, Memory Mapped Files and DLLs,
3. Process management
4. Threads and scheduling
5. Thread synchronization,
6. Inter-process Communication
7. Input/Output, Device Drivers (USB or Parallel Port),
8. File System Drivers, Filter Drivers
9. Introduction to Assembly Language, 80x86 families; program layout.
10. Data Definitions, Basic Instructions.
11. Unsigned Arithmetic; Logic and Bit Operations.
12. Modules; Separate Assembly; Argument Passing.
13. Libraries; Combining Assembly and C Code.
14. String Instructions; Arrays.
15. Macros; Structures.
16. Floating Point Instructions,
17. Bit MS-DOS.
18. BIOS Disk Accessing
19. BIOS Keyboard/Video/Graphics
20. Interrupts; TSR Programs
21. Accessing I/O Ports; 8253 Timer

Recommended Texts

1. Irvine, K. R. (2015). *Assembly language for x86 processors*. New York: Pearson Education.
2. Hart, J. M. (2010). *Windows system programming*. New York: Pearson Education.

Suggested Readings

1. Baker, A. (1996). *Windows NT device driver book: a guide for programmers, with disk with cdrom*. New York: Prentice Hall PTR.
2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6111 **Management Information Systems** **3(3+0)**

The aim of the course is to enable students to assess the opportunities and problems that managers in a wide range of organizations face. This course also helps students to understand transformational changes within and across the industries. These changes have strategic implications for many businesses. The course provides an 'Overview' of fundamental MIS concepts, using integrated Framework for 'Decision Making' and Analyzing Information Systems. The course is comprised of different types of Information Systems available for Business use in Decision Making and Business Processes systems, covering Competitive Advantage, Executive Information Systems, and Decision Support Systems. Understand basic concepts of information technology management, its application, development themes, development methodologies, development tools and technologies of Information Systems. Able to solve common business problems and produce effective solutions to business problems. Able to design a database application to solve a business problem. Participate in an organization's information systems and technology decisionmaking processes.

Contents

1. Introduction: Overview of Business Strategy Frameworks, Brief Overview of Organizational Strategies, Brief Overview of Information Systems Strategy
2. Understanding the IS Organization, what a Manager Can Expect from the IS Organization, What the IS Organization Does Not Do, IT Governance.
3. Evolution of Information Resources, Information Resources as Strategic Tools, How Can Information Resources Be Used Strategically? Strategic Alliances, Risks. Information Technology and Organizational Design, Information Technology.
4. Information Technology and the Design of Work. Information Technology
5. Information Systems Sourcing. Using Information Ethically.
6. Techniques: Project management techniques, Organizational techniques, People techniques. IT Project Development Methodologies, Managing Business Knowledge.
7. Organizational Theory. Management and Control.

Recommended Texts

1. Kroenke, D. M., Gemino, A. C., & Tingling, P. M. (2012). *Experiencing mis*. New York: Pearson Education.
2. Baltzan, P., Phillips, A. L., Lynch, K., & Blakey, P. (2008). *Business driven information systems*. New York: McGraw-Hill/Irwin.

Suggested Readings

1. Stair, R., & Reynolds, G. (2012). *Fundamentals of information systems*. Boston, MA: Cengage Learning.
2. Avison, D., & Fitzgerald, G. (2003). *Information systems development: methodologies, techniques and tools*. New York City: McGraw-Hill.

CSEC-6112 **Game Application Development** **3(3+0)**

Introduction to game development theory, framework, production and management. Learn how to create a complete computer game from start to finish. The course introduces students to learn game development and their application in the real world. Objectives include basic understanding of android development and associated technologies. Player-Centered Design, Interface & Game Feature, Interface Types, Usability. After successful completion of the course, you will master basic game development (produce, test and present a beta version of a game of your own design), understand game design and have immersed yourself in the Danish “indie” gaming community. Introduction to electronic game development and game development careers. Includes examination of history and philosophy of games, the game production process, employee factors for success in the field, and current issues and practices in the game development industry. This course covers many of the concepts involved in game design, beginning with the history of gaming and an examination of the software engineering aspects of game design. Other topics include the programming environment, game hardware, mathematical concepts, physical concepts, and graphics.

Contents

1. Building the Foundation, Historical Elements How Did We Get Here
2. Platform and Player Modes, What Is the Framework?
3. Goals and Genres? What Are the Possibilities?
4. Player Elements, Player Motivation, Geographic, Psychographics.
5. Demographics, Gender, Generation, Rating, Applying Player Market to Platform.
6. Story and Character Development: Classic Charters, Traditional Story Structure, Story Element.
7. Plot, Game Story Devices, Game Characters
8. Character Development Element, Point-of-view, Visual Character Development, Verbal Character Development, Movement.
9. Visual Character Development, Verbal Character Development, Movements, Chracter Description, Game Storytelling and Documentation
10. Gameplay: Rules to Play, Interactivity Modes, Game theory, Challenges, balance
11. Levels: Level Design, Structure, Time, Space.
12. Interface: Playcenter Design, Interface & Game Feature, Interface Types, Usability.
13. Audio: The Importance of Game Audio, Sound Effect, Voiceover, Music.

Recommended Texts

1. Meier, R. (2012). *Professional android 4 application development*. Hoboken, New Jersey: John Wiley & Sons.
2. Conway, J., & Hillegass, A. (2012). *Ios programming: the big nerd ranch guide*. Boston: Addison-Wesley Professional.

Suggested Readings

1. Hardy, B., & Phillips, B. (2013). *Android programming: The big nerd ranch guide*. Boston: Addison-Wesley Professional.
2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6113 Computer Graphics 3(3+0)

Computer graphics are an intrinsic component of many modern software applications and are often essential to the success of these applications. The objective of this course is to familiarize students with fundamental algorithms and data structures that are used in today's interactive graphics systems as well as programming and architecture of high-resolution graphics computers. The principles and practice of computer graphics are described from their mathematical foundations to the modern applications domains of scientific visualization, virtual reality, computer games and film animation. At the end of the course the students will be able to comprehend the structure of modern computer graphics systems, explain the basic principles of implementing computer graphics fundamentals , compare key algorithms for modelling and rendering graphical data, develop design and problem solving skills with applications to computer graphics and construct interactive computer graphics programs using OpenGL. This course will introduce students to all aspects of computer graphics including hardware, software and applications.

Contents

1. Fundamental Concepts: forward and backward rendering (i.e., ray-casting and rasterization),
2. Applications of computer graphics: including game engines, cad, visualization, virtual reality,
3. Polygonal representation, basic radiometry, similar triangles, and projection model, use of standard graphics APIs (see HCI GUI construction);
4. Basic rendering: rendering in nature, i.e., the emission and scattering of light and its relation to numerical integration.
5. Affine and coordinate system transformations, ray tracing, visibility and occlusion, including solutions to this problem such as depth buffering, painter's algorithm, and ray tracing
6. The forward and backward rendering equation, simple triangle rasterization, rendering with a shader-based API.
7. Texture mapping including minification and magnification (e.g., trilinear MIP-mapping), application of spatial data structures to rendering, sampling and anti-aliasing, scene graphs and the graphics pipeline.
8. Geometric modeling: basic geometric operations such as intersection calculation.,
9. Proximity tests, polynomial curves and surfaces, approximation techniques such as polynomial curves,
10. Bezier curves, spline curves and surfaces, animation as a sequence of still images.

Recommended Books

1. Hearn, D., Baker, M. P., & Carithers, W. R. (2014). *Computer graphics with opengl*. New York: Pearson Prentice Hall.
2. Gortler, S. J. (2012). *Foundations of 3D computer graphics.* Cambridge: MIT Press.

Suggested Books

1. Shreiner, D., Sellers, G., Kessenich, J., & Licea-Kane, B. (2013). *Opengl programming guide: the official guide to learning opengl*, version 4.3. Boston: Addison-Wesley.
2. James D. Foley (2013), *Computer Graphics: Principles and practice.* Boston: Addison Wesley.

CSEC-6114 **Semantic Web Techniques** **3(3+0)**

The Semantic Web is a W3C Activity for representing information in the World Wide Web in a machine-readable fashion: such that it can be used by machines not just for display purposes, but for automation, integration, and reuse across applications. This course introduces techniques that are useful stand-alone and can be integrated for building a semantic web. In this course students will be introduced to the Semantic Web vision, as well as, the languages and tools useful in Semantic Web programming. They will understand how this technology revolutionizes the World Wide Web and its uses. Ontology languages (RDF, RDF-S and OWL) and technologies (explicit metadata, ontologies, logic, and inference) will be covered. In addition, students will be exposed to; ontology engineering, application scenarios, Semantic Web Query Languages, Description Logic and state of the art Semantic Web applications, such as linked data development. Student will also learn how to develop semantic applications with Java and Jena APIs.

Contents

1. Introduction to the semantic web.
2. Structured Web Documents – XML, RDF
3. Introduction to ontologies.
4. Ontology Engineering
5. Ontology Engineering
6. Ontology languages for the semantic web.
7. Resource Description Framework (RDF).
8. Lightweight ontologies: RDF Schema.
9. Web Ontology Language (OWL).
10. Query language for RDF: SPARQL.
11. Description Logic
12. Building Semantic Web Applications (Apache Jena Framework)
13. Building Semantic Web Applications
14. Semantic Web Applications (E-learning, Web services)
15. Ontology Engineering (Protégé OWL API)
16. Semantic web and Web 2.0
17. Applications of Semantic Web.

Recommended Texts

1. Cardoso, J., Sheth, A., & Yu, L. (2009). Semantic web services, processes and applications. Berlin/Heidelberg: Springer.
2. Gortler, S. J. (2012). *Foundations of 3D computer graphics.* Cambridge: MIT Press.

Suggested Readings

1. Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). *Foundations of semantic web technologies*. Boca Raton: CRC press.

CSEC-6115 **Virtual Reality** **3 (3+0)**

Virtual reality (VR) is a powerful technology that promises to change our lives unlike any other. By artificially stimulating our senses, our bodies become tricked into accepting another version of reality. Virtual reality technology is evolving rapidly, making it undesirable to define VR in terms of specific devices that may be outdated after a few years. This course is concerned with fundamental principles that are less sensitive to particular technologies. At the end of the course the students will be able to understand fundamental techniques, processes, technologies and equipment used in virtual reality systems. Students will acquire the basic knowledge about the physiology of human vision, hearing, and perception. The conceptual and experiential possibilities of immersive VR are explored and developed in the context of a creative work or a collaborative creative work. This course enables them to understand the human interaction with virtual reality interfaces and the recent applications of virtual reality. Students will develop a research and development portfolio as a pre-production component to the creation of the creative work.

Contents

1. Definition of VR
2. Modern experiences
3. Historical perspective
4. Overview of VR systems: hardware, sensors, displays, software, virtual world generator
5. Human senses
6. Perceptual psychology
7. Psychophysics
8. The geometry of virtual worlds: geometric modeling and transformations
9. Light propagation, lenses and images, diopters, spherical aberrations, optical distortion
10. Lens aberrations, Spectral properties
11. The eye as an optical system, cameras, visual displays
12. Physiology of human vision
13. Visual perception: depth, motion, section, color
14. Combining information from multiple cues and senses
15. Visual rendering: graphical rendering, ray tracing, shading, BRDFs, rasterization
16. VR rendering problems, Anti-aliasing, Distortion shading
17. Image warping, Motion in real and virtual worlds
18. Tracking: tracking systems, estimating rotation, IMU integration, and drift errors
19. Evaluating VR systems and experiences
20. Touch, haptics, taste, and smell
21. Robotic interfaces, telepresence
22. Brain-machine interfaces

Recommended Texts

1. Steven, M. L. (2019). *Virtual reality*. Cambridge: Cambridge University Press.

Suggested Readings

1. Shirley, P., Marschner, S. & Ashikhmin, M. (2009). *Fundamentals of computer graphics*. Natick, Mass: A K Peters.

CSEC-6116 Mobile Computing 3(3+0)

The course is aimed to prepare students to grasp the concepts and features of mobile computing technologies and applications. This course will provide graduate students of Information Systems with both broad and in-depth knowledge, and a critical understanding of mobile computing from different viewpoints: infrastructures, principles and theories, technologies, and applications in different domains. The course will provide a complete overview of the mobile computing subject area, including the latest research. In Unit 6, each student will have the opportunity to delve into more specific technology and/or application domains by forming a small special interest group (SIG) with their fellow students. In addition, through presentations, Q&A, and debates, students will have the opportunity to further explore specific topics. This course will provide broad and in-depth knowledge, and a critical understanding of mobile computing from different viewpoints: infrastructures, principles and theories, technologies, and applications in different domains. The course will provide a complete overview of the mobile computing subject area, including the latest research.

Contents

1. Introduction to Mobile Computing
2. Creating Consumable Web Services for Mobile Devices
3. Memory Management
4. Mobile Applications
5. Mobile User-Interface Design
6. Dynamic Linking
7. Concurrency
8. Managing Resources
9. Security
10. Introduction to Mobile Application Development with Android
11. Introduction to Mobile Application Development with IOS
12. Introduction to Mobile Application Development with Windows Phone
13. Introduction to Mobile Application Development with Blackberry

Recommended Texts

1. Mikkonen, T. (2007). *Programming mobile devices: an introduction for practitioners*. Hoboken, New Jersey: John Wiley & Sons.
2. McWherter, J., & Gowell, S. (2012). *Professional mobile application development*. Hoboken, New Jersey: John Wiley & Sons.

Suggested Readings

1. B'far, R. (2004). *Mobile computing principles: designing and developing mobile applications with UML and XML*. Cambridge: Cambridge University Press.
2. Fling, B. (2009). *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps*. Champaign: O'Reilly Media, Inc.
3. Adelstein, F., Gupta, S. K., Richard, G., & Schwiebert, L. (2005). *Fundamentals of mobile and pervasive computing* (Vol. 1). New York: McGraw-Hill.
4. Neil, T. (2014). *Mobile design pattern gallery: UI patterns for smartphone apps*. Champaig: O'Reilly Media, Inc.

CSEC-6117 **Data Warehousing** **3(3+0)**

This course gives an introduction to methods and theory for development of data warehouses and data analysis using data mining. Data quality and methods and techniques for preprocessing of data. Modeling and design of data warehouses. Algorithms for classification, clustering and association rule analysis. Practical use of software for data analysis. To learn concepts and skills for designing data warehouses and creating data integration workflows. Introduction of tools for data warehousing. This course is covered to understand the basic definitions and concepts of data warehouses, understand data warehousing architectures and to describe the processes used in developing and managing data warehouses. This will explain data warehousing operations and explain the role of data warehouses in decision support. Students will be able to understand complex variable theory, application of harmonic conjugate to get orthogonal trajectories and analytic function. Plot the image of the curve by a complex transformation from z-plane to w-plane.

Contents

1. Introduction to Data Warehousing: Data Staging and ETL, Multidimensional Model, Meta-data, Accessing Data Warehouse, ROLAP, MOLAP, and HOLAP.
2. Data Warehouse System Lifecycle: Risk Factors, Top-Down vs Bottom-Up, Data Mart Design Phases, Methodological Framework Testing Data Marts.
3. Analysis and Reconciliation of Data Sources: Inspecting and Normalization Schemata, Integration Problems, Integration Phases, Defining Mapping.
4. User Requirement Analysis: Interviews, Glossary-based Requirement Analysis
5. Dimensional Fact Model, Events and Aggregation, Temporal Aspects, Overlapping Fact Schemata, Formalizing the Dimensional Fact Model.
6. Conceptual Design: ER Schema-based Design, Relational Schema-based Design, XML Schema-based Design, Mixed-approach Design. Requirement-driven Approach Design.
7. Workload and Data Volume
8. Logical Modeling: MOLAP and HOLAP Systems, ROLAP Systems, Views, Temporal Scenarios.
9. Logical Design: From Fact Schemata to Start Schemata, View Materialization, View Fragmentation.
10. Data-staging Design: Population Reconciled Databases, Cleansing Data, Populating Dimensional Tables, Populating Fact Tables, Populating Materialized View
11. Indexes for the Data Warehouse: B*-Tree Indexes, Bitmap Indexes, Projection Indexes, Join & Star Indexes, Spatial Indexes, Join-Algorithm.
12. Physical Design: Optimizers, Index Selection, splitting a Database into Tablespaces

Recommended Texts

1. Inmon, W. H. (2000*). Building the data warehouse*. Hoboken, New Jersey: John Wiley & Sons.

Suggested Readings

1. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., & Becker, B. (2008). *The data warehouse lifecycle toolkit*. Hoboken, New Jersey: John Wiley & Sons.
2. Ponniah, P. (2011). *Data warehousing fundamentals for IT professionals*. Hoboken, New Jersey: John Wiley & Sons.

CSEC-6118 **Data Mining** **3 (3+0)**

The course introduces students with basic applications, concepts, and techniques of data mining and to develop their skills for using recent data mining software to solve practical problems in a variety of disciplines. At the end of the course the students will be able to understand the applications, concepts, and techniques of data mining. Learn different data mining tools and apply basic data mining techniques to actual problems. Moreover, student will be able to identify appropriate data mining algorithms to solve real world problems and Compare and evaluate different data mining techniques like classification, prediction, clustering and association rule mining and web mining. The general objective of data mining is to look for patterns in data. One example of this might be where we consider data to be the number of questions on Quora. Then one such pattern might be the number of such questions asking about the objective of data mining.

Contents

1. Data-Mining Concepts, Data-Mining Process, Large Data Sets, Data Warehouses for Data Mining, Business Aspects Data Mining.
2. Preparing the Data: Raw Data- Representation, Characteristics, Transformation; Missing Data, Time-Dependent Data, Outlier Analysis.
3. Data Reduction: Dimensions of Large Data Sets, Feature Reduction, Relief Algorithm, Entropy Measure for Ranking Features, PCA, Value Reduction, Feature Discretization
4. Learning Machine, SLT, Types of Learning Methods, SVMs, kNN: Nearest Neighbor Classifier, Model Selection versus Generalization, Model Estimation.
5. Statistical Inference, Assessing Differences in Data Sets, Bayesian Inference, Predictive Regression, ANOVA, Logistic Regression, Log-Linear Models, LDA.
6. Decision Trees and Decision Rules: Generating & Pruning Decision Tree, CART Algorithm & Gini Index, Limitations of Decision Trees and Decision Rules.
7. Artificial Neural Networks
8. Ensemble Learning
9. Clustering, Similarity Measures, Agglomerative Hierarchical Clustering, Partitional Clustering, Incremental Clustering, DBSCAN Algorithm.
10. Association Rules
11. Web Mining and Text Mining
12. Genetic Algorithms, Fuzzy Sets and Fuzzy Logic

Recommended Texts

1. Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. Piscataway, New Jersey Hoboken, NJ: IEEE Press Wiley.
2. Han, J., Kamber, M. & Pei, J. (2012). *Data mining: concepts and techniques.* Amsterdam Boston: Elsevier/Morgan Kaufmann.

Suggested Readings

1. Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.
2. Singh, R. & Asthana, A. (2012). *Data mining and data warehousing practical machine learning*

tools techniques. Saarbrücken: LAP LAMBERT Academic Publishing.

CSEC-6119 Business Intelligence and Analytics 3 (3+0)

Business Intelligence (BI) is the field of combining data, technology, business processes, and analytics to optimize business decisions and drive success. The ability to generate insights using data in today's world is crucial for any organization's success into the future. Turning data into valuable information is a necessity for ever-changing markets. This course will give an introduction to analytical tools and skills that can be used to understand, analyses, and evaluate the challenges and opportunities for an organization. Students will learn about various technical aspects of BI and understand the processes involving in planning, designing, building, and maintaining the BI environment. Further, this course will develop skills in the use and application of various techniques and tools for driving insights from data for effective business decision making. This course introduced to key data analytics concepts such as systems thinking, multi-level perspectives and multidisciplinary methods for envisioning futures, and apply them to specific real-world challenges you and your organization may face.

Contents

1. Value drivers, performance metrics and key performance indicators
2. Use cases for BI
3. BI success factors, strategic versus tactical planning, BI strategy and plan
4. BI environment: analytics platform, frameworks, services, and systems evaluation
5. Business process and information flow
6. Data requirements analysis
7. Data warehouses and the technical BI architecture
8. Data profiling
9. Business rules
10. Data quality
11. Data integration
12. Deriving insight from data
13. Knowledge discovery & delivery
14. Installations, configuring and maintaining the BI server
15. Creating reports using answers and dashboards.

Recommended Texts

1. Rittman, M. (2013). *Oracle business intelligence 11g developer's guide*. New York: McGraw-Hill.

Suggested Readings

1. Larson, B. (2012). *Delivering business intelligence with microsoft sql server 2012*. New York: McGraw-Hill.
2. Vitt, E., Luckevich, M. & Misner, S. (2002). *Business intelligence: making better decisions faster*. Redmond, Wash: Microsoft Press.
3. Laberge, R. (2011). *The data warehouse mentor: Practical data warehouse and business intelligence insights*. New York City: McGraw-Hill.

CSEC-6120 Database Administration & Management 3 (3+0)

This course is designed for students to gain a comprehensive overview of all the skills necessary to become successful database administrators. This course provides an understanding of the internal functionality of the database management system by Oracle corporation. Students are required to have a working knowledge of the relational database model as well as PL/SQL programming skills. Students will learn how to administrator databases implemented in Oracle DBMS. They will learn how to configure and install DBMS, perform common administrative tasks using GUI, and know how to use the SQL language in order to manipulate data and DBMS. In addition to practicing existing recovery, backup, and network plans, students will gain hands-on experience practicing database-related procedures such as how to monitor a database and manage its performance. Students will learn how to administrator databases implemented in Oracle DBMS. They will learn how to install and configure DBMS, perform common administrative tasks using GUI.

Contents

1. Oracle installation and Oracle architectural components
2. Oracle physical structures
3. Managing an Oracle Instance
4. Creating database and data dictionary
5. Managing control files and redo log files
6. Managing tablespaces, data files, segments, and blocks
7. Managing undo data
8. Indexes and privileges
9. Maintaining data integrity and constraints
10. Basic Oracle Net Architecture
11. Server-side configuration
12. Client-side configuration
13. Backup and recovery
14. Rollback segments
15. Latches
16. Rollback
17. Shared servers
18. Locks
19. Block efficiency
20. Statistics and monitoring index usage

Recommended Texts

1. Bryla, B. & Loney, K. (2008). *Oracle database 11g DBA handbook*. New York: McGraw-Hill.
2. Mullins, C. (2013). *Database administration: the complete guide to dba practices and procedures*. New York: Addison-Wesley.

Suggested Readings

1. Connolly, T. & Begg, C. (2010). *Database systems: a practical approach to design, implementation, and management*. Boston: Addison-Wesley.
2. Loney, K. (2009). *Oracle database 11g: the complete reference*. New York: McGraw-Hill.

CSEC-6121 Advance Database Management 3(3+0)

This course will address the advanced issues in modern database systems and applications. Databases underlie most complex computing systems. Software systems that involve complex databases are heterogeneous. Simple approaches designed for small, centralized, homogeneous databases are ineffective and inappropriate for dealing with large, distributed, heterogeneous environments. Data-related issues in building, analyzing, and maintaining complex software systems are the focus of this course. Techniques for operating and maintaining heterogeneous database systems, business problem-solving, and decision-support systems are also discussed. At the end of the course, the students will learn about transaction processing, concurrency control, crash recovery, and database security. Furthermore, this course also provides an overview of data warehousing, OLAP, and data mining Explain and evaluate the fundamental theories and requirements that influence the design of modern database systems. Assess and apply database functions and packages suitable for enterprise database development and database management. Critically evaluate alternative designs and architectures for databases and data warehouses.

Contents

1. PL/SQL: introduction to PL
2. Transaction-processing monitors
3. Transactional workflows
4. Main-memory databases
5. Real-time transaction systems and long-duration transactions
6. Transaction management in multi-databases
7. Concurrency control: locks
8. Optimistic concurrency control and timestamping concurrency control.
9. Object-based databases and xml object-based databases
10. Programming languages: OO vs OR
11. XML: structure, document schema, querying, API, and applications
12. Data warehousing and its design
13. OLAP and data mining
14. Database security

Recommended Texts

1. Silberschatz, A., Korth, H. & Sudarshan, S. (2011). *Database system concepts*. New York: McGraw-Hill.

Suggested Readings

1. Connolly, T. & Begg, C. (2010). *Database systems: a practical approach to design, implementation, and management*. Boston: Addison-Wesley.
2. Bayross, I. (2002). *SQL, Pl/SQL the programming language of Oracle*. New Delhi: BPB Publications.
3. Han, J., Kamber, M. & Pei, J. (2012). *Data mining: concepts and techniques*. Amsterdam Boston: Elsevier/Morgan Kaufmann.

4. Gertz, M. & Jajodia, S. (2008). *Handbook of database security: applications and trends*. New York: Springer.

CSEC-6122 Business Process Management 3 (3+0)

This course looks at ways in which business processes can be analyzed, redesigned, and improved thus ensuring that they are meeting the needs of customers and the enterprise. A business process is a set of related activities that together realize a business goal in an organizational and technical context. These processes take place in a single organization but may need to interact with processes in other organizations. This course will introduce you to business process management. You'll learn how business processes can help you improve your company's bottom line by providing a higher level of quality and consistency for your customers. Business Process Management (BPM) is concerned with the concepts, methods, and techniques that support the design, improvement, management, configuration, enactment, and analysis of business processes that deliver lean and customer focused business processes. BPM includes process modelling that includes defining, analyzing, and improving processes. Students will be able to understand business process from a management and process analyst perspective, learn skills, analytical frameworks and general principles for managing business processes.

Contents

1. Business introduction
2. Evolution of enterprise systems architectures
3. Business process modeling
4. Process orchestrations
5. Process choreographies
6. Properties of business processes
7. Business process management architectures
8. Business process management methodology

Recommended Texts

1. Weske, M. (2012). *Business process management*: concepts, languages, architectures. Berlin New York: Springer.

Suggested Readings

1. Yvonne, L. A., Martin, B., Tony, B., Bruce, D. D., Jason, F., Daniel, J. M., ... & Robyn, L. R. (2009). *Business Process Management Common Body of Knowledge*. Chicago: Association of Business Process Management Professionals.
2. Becker, J., Kugeler, M. & Rosemann, M. (2011). *Process management: a guide for the design of business processes*. Berlin London: Springer.
3. Jeston, J. & Nelis, J. (2008). *Business process management: practical guidelines to successful implementations*. Amsterdam Boston London: Elsevier/Butterworth-Heinemann.
4. Malik, T. (2009). *Process management: practical guidelines to successful implementations*. New Delhi: Global India Publications.

CSEC-6123 **Knowledge Management** **3 (3+0)**

Knowledge, knowledge products, and knowledge processes are the key ingredients of productivity and profitably in the business world. Knowledge revolution has given birth to knowledge economies. Knowledge management (KM) is the field of managing human knowledge. Knowledge Management is commonly associated with processes like knowledge creation, knowledge sharing, knowledge storage, knowledge refinement, etc. Knowledge management (KM) is an area that has captured the attention of many organizations that are concerned with the way's knowledge is managed more effectively. KM offers systematic methods in leveraging and managing organizational knowledge through KM processes of creation, storing, sharing, and application of knowledge. The need is to gain a sustainable competitive edge among partners as well as competitors by learning how to leverage intangible assets in new and creative ways. This course introduces students to appraise current thoughts on knowledge management in the light of contemporary debates on knowledge productivity, strategic capability, and organizational learning. Further, it enables them to learn how to develop, manage, and evaluate knowledge management theories, models, frameworks, systems, initiatives, and best practices.

Contents

1. History and paradigms of knowledge management
2. Types of knowledge: explicit knowledge, tacit knowledge, embedded knowledge
3. KM processes: knowledge discovery/ detection
4. KM frameworks and models: SECI, Alen Frost's, and Van Buren's models
5. Knowledge capture and codification: group, knowledge codification
6. Knowledge sharing and communities of practice: types of communities
7. Knowledge application: task analysis and modeling, knowledge reuse
8. The role of organizational culture: different types of cultures
9. Knowledge management tools: knowledge blogs, mashups, PKM
10. Knowledge management strategy: knowledge audit, gap analysis
11. The value of knowledge management: ROI and metrics
12. Organizational learning
13. Organizational memory
14. Major categories of knowledge management roles
15. The profession and ethics of knowledge management

Recommended Texts

1. Dalkir, K. (2017). *Knowledge management in theory and practice*. Cambridge: Massachusetts Institute of Technology.
2. Pasha, M. & Pasha, S. (2012). *Essentials of knowledge management: concepts, theories and practices.* Hoboken, New Jersey: John Wiley & Sons.

Suggested Readings

1. Tiwana, A. (2007). *The knowledge management toolkit: orchestrating It, strategy, and knowledge platforms.* New York: Prentice Hall PTR.
2. Geisler, E. & Wickramasinghe, N. (2009). *Principles of knowledge management: theory, practices, and cases*. Armonk, N.Y: M.E. Sharpe.

CSEC-6124 **Graph Theory & Algorithms** **4(3+1)**

This course is aimed to cover a variety of different problems in Graph Theory. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Various graphs algorithms will also be taught along with its analysis. By taking this course, one would be able to master fundamental concepts in Graph Theory, get to know a wide range of different Graphs, and their properties, be able to perform Elementary, Advanced Operations on Graphs to produce a new Graph, understand Graph Coloring, understand Eulerian and Hamiltonian paths and circuits. And many related topics to Paths, know how to turn a Graph into a Matrix and vice versa, obtain a solid foundation in Trees, Tree Traversals, and Expression Trees, have a good understanding of Graph Match. Topics covered in this course include: Fundamental Concepts of Graphs, Sub-Graphs and Super-Graphs, Connected Graphs (Walks and connection), Trees (including forests, spanning trees, Calay's Formula for positive integers, and other applications of trees), Non-Separable graphs and its applications, Connectivity among graphs, Planar graphs, Vertex coloring, edge coloring, Hamiltonian Cycles (including both Hamiltonian and non-Hamiltonian Graphs), Eigen values of graphs, Covering and packings.

Contents

1. Fundamental Concepts of Graphs: What is A Graph, Simple Graphs, Graph and Their Representations, Isomorphism and Automorphisms,
2. Labelled Graphs, Graphs Arising from Other Structures, Incidents Graphs, Union and Intersection Graphs, Cartesian Product.
3. Directed Graphs.
4. Sub-Graphs: Sub-Graphs and Super graphs, Spanning and Induced Sub-Graphs, Decomposition and Coverings, Edge Cuts and Bonds, Even Sub-Graphs, Graph Reconstruction.
5. Connected Graphs: Walks and Connection, Cut Edges, Connection to Diagraphs
6. Trees: Forests and Trees, Spanning Tree, Clay's Formula, Fundamental Cycles and Bonds, Co-Tree, Trees and Distance. Applications of Tree.
7. Non-separable Graphs: Cut Vertices, Separations and Blocks, Ear Decompositions, Strong Orientations, Directed Ear Decompositions, Even Cycles Decompositions.
8. Connectivity: Vertex Connectivity, Fan Lemma, Edge Connectivity, Three-Connected Graphs, Sub modularity, Determining, Chordal Graphs, Simplicial vertices.
9. Planar Graphs: Plane and Planar Graphs, Duality, Euler's Formula, Kuratowski's Theorem.
10. Vertex Colorings: Chromatic Numbers, Critical Graphs, Girt hand Chromatic Number

Recommended Texts

1. Beezer, R. A. (2008). *Graph theory,* by JA Bondy and USR Murty. Berlin: Springer.
2. Chartrand, G., & Zhang, P. (2013). *A first course in graph theory.* North Chelmsford: Courier Corporation.

Suggested Readings

1. Chartrand, G., Lesniak, L., & Zhang, P. (2010). *Graphs & digraphs* (Vol. 39). Boca Raton: CRC press.
2. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

CSEC-6125 Natural Language Processing 3 (3+0)

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes
4. Context-free parsing, Earley algorithm
5. Extending CFG.
6. Probabilistic parsing, Parsing tricks, Human sentence processing.
7. Semantics, Forward-backward algorithm
8. Expectation Maximization.
9. Finite-state algebra
10. Finite-state implementation
11. Finite-state tagging
12. Noisy channels and FSTs, More FST examples.
13. Programming with regexps, Morphology and phonology.
14. Optimal paths in graphs, Structured prediction.
15. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing*. New York: Prentice Hall
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. Massachusetts: OReilly Media.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Bengio, Y. (2007). Learning deep architectures for AI (Technical Report 1312). *Statistical methods for speech recognition*. Cambridge: MIT press.

CSEC-6126 **Grammar Engineering** **3 (3+0)**

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview, LFG Basics.
3. LFG Basics II, Templates I, MacOSX, Unix.
4. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file
5. Lexical Rules, Passive and Argument alternations.
6. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.
11. Finite-State Morphology (FSM) I.
12. FSM II (-unknown), Free Word Order and the Shuffle Operator.
13. Meta-categories, Meta-rule-macros and Coordination.
14. Project

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. Redwood: Publications Stanford.
2. Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., & Newman, P. (2008). *XLE documentation.* Coyote: Palo Alto Research Center.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar*. Leiden: Brill.
2. Dalrymple, M., Kaplan, R. M., *Formal issues in lexical-functional grammar*. NY: Maxwell.

CSEC-6127 Text Processing & Data Analysis 3 (3+0)

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions
4. Writing Safe Code.
5. Control Structures
6. File Input / Output.
7. Introduction to Text Processing
8. Text Processing Functions.
9. Loop Control
10. Hashes
11. DBM Databases
12. Advanced Sorting.
13. Regular Expressions, Environment Variables, CGI-Programming.
14. Process Management, References and Data Structures.
15. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. Champaign: O'Reilly & Associates, Inc.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming Perl: Unmatched power for text processing and scripting*. Champaign: O’Reilly Media, Inc.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl Cookbook: solutions & examples for perl programmers*. Champaign, IL 61820, United States: O'Reilly Media, Inc.
2. Lidie, S., & Walsh, N. (2002). *Mastering Perl/Tk: graphical user interfaces in perl*. Champaign: O'Reilly Media, Inc.

CSEC-6128 Speech Recognition & Synthesis 3 (3+0)

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background: N-gram Language Modeling
2. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization, Phonetics Speech Synthesis, pages 1-10 , Optional Advanced Reading, Text Segmentation and Organisation, Text Decoding.
3. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software, Prosody Prediction from Text.
4. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis), Unit Selection Synthesis, Optional Advanced Reading.
5. ASR: Noisy Channel Model, Bayes, HMMs, Forward, Viterbi, Hidden Markov Models, Automatic Speech Recognition.
6. ASR: Feature Extraction and Acoustic Modeling, Evaluation, Speech Recognition: Advanced Topics.

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing*. New York**:** Prentice Hall
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, Massachusetts: MIT press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-418 Introduction to Computational Linguistics 3 (3+0)

This course provides a general introduction to contemporary ideas in Computational Linguistics (Natural Language Processing, Human Language Technology). The objective of this course is to give a theoretically grounded introduction to contemporary work in Computational Linguistics and to introduce standard methods for processing words (morphology). This course is covered to introduce standard methods for sentence processing (parsing and generation) and to introduce and develop an understanding of some key computational notions. Fundamental concepts of automata theory and formal languages to form basic models of computation which provide foundation of many branches of computer science, e.g. compilers, software engineering, concurrent systems. The prospective graduate will have acquired insight into the various computational research methods used in linguistics. Furthermore, they will have developed an analytical skillset which is indispensable in the interdisciplinary study of natural language. The student is aware of, and adheres to, the highest ethical standards in science, values integrity above self-interest, and aspires to academic excellence.

Contents

1. Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms, Language, Thought, and Understanding the Near-Term Future,
2. Regular Expressions, Finite-State Automata, Regular Languages and FSAs
3. Morphology and Finite-State Transducers
4. Computational Phonology and Text-to-Speech, Speech Sounds and Phonetic Transcription
5. Probabilistic Models of Pronunciation and Spelling, Dealing with Spelling Errors, Detecting Non-Word Errors, Probabilistic Models, Applying the Bayesian method to spelling
6. N-grams, Counting Words in Corpora, Simple N-grams, Smoothing, Backoff, Deleted Interpolation, N-grams for Spelling and Pronunciation, Entropy
7. HMMs and Speech Recognition, Hidden Markov Models, The Viterbi Algorithm, Advanced Methods for Decoding, Acoustic Processing of Speech
8. Word Classes and Part-of-Speech Tagging, English Word Classes, Tagsets for English, Part of Speech Tagging, Rule-based Part-of-speech Tagging, Context-Free Grammars for English
9. Parsing with context free grammars, parsing as search
10. Features and Unifications, unification of feature structures, features structures in grammar, implementing unification, parsing with unification constraints

Recommended Books

1. Daniel Jurafsky and James H. Martin. (2008) *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition.* New York: Prentice Hall.
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge: MIT press

Suggested Readings

1. Philipp Koehn. (2012) *Statistical machine translation*. Cambridge: Cambridge University Press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CMPC-306 **Human Computer Interaction** **3 (3+0)**

This course will equip the students with the tools and techniques required to design user-friendly interactive systems, latest theories, principles guidelines in Human-Computer Interaction (HCI), User Experience (UX), usability and Interaction Design, Moreover the students will be able to understand the entire user-centered design and evaluation process; from understanding user needs to design interactive systems that meet those needs, to evaluate the usability of those (and existing systems) through user research. The students will be able to explain the capabilities of both humans and computers from the viewpoint of human information processing. Students will also have an idea about understanding users' problems, designing usable user interfaces and evaluating these user interfaces and UX with end users. Design an interactive application, applying a user-centered design cycle and related tools and techniques (e.g., prototyping), aiming at usability and relevant user experience. Usability inspection methods, Usability testing methods, New Interaction Technologies, Ubiquitous computing and augmented realities.

Contents

1. The human: Input–output channels, Human memory, Physical capabilities, Thinking, Emotion, Psychology and the design of interactive systems.
2. The interaction, Frameworks and HCI, Ergonomics, Interaction styles, Elements of the WIMP interface, Interactivity, Experience, engagement and fun., Principles of GUI
3. Interaction design basics, Interaction types, User focus, Scenarios, Navigation design, Screen design and layout, Iteration and prototyping. considering assistive technologies
4. User and task analysis, uses of task analysis, accessibility, standards, User experience design.
5. Design rules, Standards, benchmarks, Guidelines, Golden rules and heuristics, HCI patterns.
6. Evaluation techniques and Goals, Evaluation through expert analysis for evaluation of user-centered design, articulate evaluation criteria and compliance to relevant standards
7. Universal design principles, Multi-modal interaction, designing for diversity.
8. User support, approaches to user support, designing user support systems.
9. Groupware systems, Computer-mediated communication, Meeting and decision support systems, Shared applications and artifacts, Frameworks for groupware.
10. Ubiquitous computing and augmented realities, Virtual and augmented reality, Information and data visualization.

Recommended Texts

1. Preece, J., Rogers, Y., & Sharp, H. (2019). *Interaction design: Beyond human-computer interaction* (5th ed.) Hoboken, New Jersey: John Wiley & Sons Ltd.
2. Dix, A., Finlay, J., Abowd, G.D., & Beale, R. (2004). *Human computer interaction.* New York: Prentice Hall.

Suggested Readings

1. *Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). About face: the essentials of interaction design. Hoboken NJ: John Wiley & Sons.*
2. *Benyon, D (2013) Designing interactive systems: A comprehensive guide to HCI, UX and interaction. Noida, Uttar Pradesh: Pearson.*

BS
INFORMATION TECHNOLOGY

CMPC-5201 **Programming Fundamentals** **4(3+1)**

In this course the student will gain a broad understanding of modern computer programming. The student will acquire introductory skills in problem analysis, solution design, and program construction. Through practical programming activities, the student will gain an appreciation of the nature and history of computer programming. Develop, understand, test, and evolve substantial programs using a modern IDE, and associated configuration tools. The significant philosophies and logical programming, including models for I/O, processing, and all related terminology will be taught. Simple programs will be constructed, using a number of different logical, calculation and algorithm. Upon successful completion of this course, the student will have reliably demonstrated the ability to solve basic programming problems using a variety of skills and strategies. Use pseudo-code and visual modeling to prepare clear and accurate program documentation and models. Examine working programs to identify their structures. Students will learn to apply appropriate techniques to create entry-level programs from models.

Contents

1. Overview of Computer Programming
2. Principles of Structured and Modular Programming: Algorithms, Pseudo code, flowchart representation. Basic Data Types
3. Unary and Binary (arithmetic, relational, arithmetic assignment) operators. Arithmetic (Expression) in C, C Programming Basics,
4. Decision Statements: if statement, if-else statement, Multi if-else-if statement. Nested if-else statements, Switch Statement.
5. Decision Statements: Conditional operator, Logical Operators. Program Control: Repetition Essentials, Counter-Controlled Repetition.
6. Loops: while loop, do while loop.
7. Nested loop structures, Control Statements, break and continue Statements, Logical Operators. Programming Practices.
8. Arrays, Searching techniques, Sorting Arrays: selection sort, bubble sort.
9. Strings: String Library Functions Characters and Strings
10. Function declaration, definition, Passing Arguments to functions, Returning values from functions. Arguments pass by reference and pass by copy.
11. Functions: Passing arrays and strings to functions. Inline functions, Default arguments, Local and global variables,
12. Pointers and their purpose. Pointer expressions. Pointers and arrays, Pointers in functions.
13. Static and dynamic memory allocation, Memory Management using Pointers. Problems with pointers
14. Defining structures, Initializing Structures, Accessing Structure Members. Passing Structures to functions, Structures using pointers.
15. File Processing
16. Revise + Lab Exam discussion, Class discussion

Recommended Texts

1. Deitel, P. J., & Deitel, H. M. (2008). *C++ how to program*. New York: Pearson Prentice Hall.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Kochan, S. G. (2015). *Programming in c.* New York: Pearson education.
2. Deitel, P., & Deitel, H. (2011). *Java how to program*. New York: Prentice Hall Press.

CMPC-5202 Object Oriented Programming 4(3+1)

This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low-level language features. The course briefly covers the mapping of UML design to C++ implementation and object-oriented considerations for software design and reuse. The course also relates C++ to GUI, databases, and real-time programming. The course material embraces the C++ language standard with numerous examples demonstrating the benefits of C++. Develop an understanding of object-oriented design artifacts and their mapping to object-oriented programming. Design and implement object-oriented solutions for small systems involving single/multiple objects. On completion of the course the students should be able to describe the meaning of the object-oriented paradigm, and create class hierarchies using the object-oriented design process. Design and implement C++ programs for complex problems, making good use of the features of the language such as classes, inheritance and templates.

Contents

1. Objects and Classes, Abstraction, Encapsulation.
2. Final Classes, Nested and Inner Classes.
3. Inheritance, Abstract Classes, Concrete Classes, Inheritance and Encapsulation.
4. The is-a Relationship, Inheritance via Abstract Classes, Extending the Hierarchy, Up casting and Down casting, Interfaces.
5. Composition, the has-a Relationship.
6. Polymorphism.
7. Polymorphism, Dynamic (or Late) Binding.
8. Interfaces and Polymorphism.
9. The Wrapper Classes, Boxing and Un-Boxing, Packages.
10. Exceptions and Exception Handling.
11. File Systems and Paths, File and Directory Handling and Manipulation, Input/output Streams, Reading Binary Data, writing binary Data, Writing Text (Characters), Reading Text (Characters), Logging with Print Stream, Random Access Files, Object Serialization.
12. Collections, for-each Loop.
13. GUI Concepts, Components and Containers, Abstract Windows Toolkit and Swing, Windows and Frames, Layout Managers, Panels.
14. Event-Driven Programming, The delegation Event Model.
15. Event Classes, Mouse Events, Keyboard Events, Using Actions.
16. Component and J Component, Buttons, Labels, Text Fields, Text Areas, Dialog Boxes, Checkboxes and Radio Buttons, Menus, J-Slider, J-Tabbed Pane.

Recommended Texts

1. Ralph Bravaco and Shai Simonson. (2010). *Java programming: from the ground up*. New York: McGraw-Hill Higher Education.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Horton, I. (2011). *Ivor horton's beginning java* (7th ed.). New York: John Wiley & Sons.

2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CMPC-5101 Discrete Structures 3(3+0)

The course provides a solid theoretical foundation of discrete structures as they apply to Computer Science problems and structures. Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic. Express a logic sentence in terms of predicates, quantifiers, and logical connectives Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion the students will learn how to use mathematical notation and solve problems using mathematical tools. On completion of the module the student should Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving, Understand the basics of discrete probability and number theory, and be able to apply the methods from these subjects in problem solving, Be able to use effectively algebraic techniques to analyse basic discrete structures and algorithms, Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic example, understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

Contents

1. Logic: Propositional Equivalences
2. Predicates and Quantifiers, Nested Quantifiers
3. Methods of Proof.
4. Sets & Functions, Sequences and Summations.
5. Algorithms: The Growth of Functions, Complexity of Algorithms, the Integers and Division,
6. Matrices.
7. Number Theory and Cryptography.
8. Advanced Counting Techniques: Recurrence Relations, Solving Recurrence Relations, Divide-and-Conquer Algorithms
9. Recurrence Relations, Generating Functions
10. Inclusion-Exclusion & its Application.
11. Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations,
12. Closures of Relations, Equivalence Relations, Partial Orderings.
13. Graph: Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths
14. Shortest-Path Problems, Planar Graphs, Graph Coloring.
15. Trees: Applications of Trees, Tree Traversal
16. Spanning Trees, Minimum Spanning Trees.

Recommended Texts

1. Rosen, K. H., & Krithivasan, K. (2012). *Discrete mathematics and its applications: with combinatorics and graph theory* (7^{th} ed.). New Delhi: Tata McGraw-Hill Education.
2. R. Johnsonbaugh. (2008). *Discrete mathematics* (7^{th} ed.). New York: Pearson

Suggested Readings

1. Maurer, S. B., & Ralston, A. (2004). *Discrete algorithmic mathematics*. New York: AK Peters. Wellesley, MA.
2. Kolman, B., Busby, R. C., & Ross, S. C. (2003). *Discrete mathematical structures*. New Delhi:

Prentice-Hall, Inc.

CMPC-5203 Data Structure and Algorithms 4(3+1)

This course provides in-depth coverage theory, practice and methods of data structures and algorithm design. Explain the layered approach that makes design, implementation and operation and identify aspect of complex operating system. Course define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms, min-max algorithm. Students will be able to assess how the choice of data structures and algorithm design methods impacts the performance of programs. Choose the appropriate data structure and algorithm design method for a specified application. Write programs using object-oriented design principles. Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, tournament trees, binary search trees, and graphs and writing programs for these solutions. Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound and writing programs for these solutions.

Contents

1. Introduction to Data Structure, primitive java, Reference Types, Algorithm Analysis.
2. Java collections API (The Java Collections Framework.
3. Recursion, Sorting Algorithms (Bubble Sort, Selection Sort, Insertion Sort, Shell Sort).
4. Sorting Algorithms (Merge Sort, Quick Sort, Heap Sort, Speed Limit for comparison Sorts, Radix Sort, Bucket Sort), Randomization.
5. Stack and Queue.
6. Linked Lists, Hash Table, Trees.
7. Binary Search Trees, Priority Queue: The Binary Heap.
8. Splay Trees, Merging Priority Queues.
9. Graphs (Simple Graphs, Graph Terminology, Paths and Cycles, Isomorphic Graphs, the Adjacency Matrix for a Graph, the Incidence Matrix for a Graph, the Adjacency List for a Graph, Digraphs
10. Graphs (Paths in a Digraph, Weighted Digraphs and Graphs, Euler Paths and Hamilto-nian Cycles, Dijkstra's Algorithm, Graph Traversal Algorithms
11. Data Structure Applications (Balanced-Symbol Checker, A Simple Calculator, File Compression, A Cross-reference Generator, The Josephus problem, Event-Driven Simulation.

Pre-Requisite: Object Oriented Programming

Recommended Texts

1. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). *Data structures and algorithms in C++*. Hoboken: John Wiley & Sons, Inc
2. VVEISS, M. A. (2002). *Data structures and problem solving using Java*. New York: Pearson Education, Inc.

Suggested Readings

1. Hubbard, J. R., & Huray, A. (2004). *Data structures with java*. New York: Pearson Prentice Hall.
2. Carrano, F. M., & Savitch, W. J. (2003). *Data structu1qres and abstractions with java*. New

York: Prentice Hall.

CMPC-5204 **Operating Systems** **4(3+1)**

This course describes the general understanding of the principles and concepts governing the functions of operating systems. Also, this course explains the layered approach that makes design, implementation and operation. It also helps in identifying aspect of complex operating system. Students have the logical, algorithmic, and mathematical capability to model and analyze real-world problems in different application domains, to devise the problem-solving schemes accordingly, and to validate the correctness and effectiveness of the schemes. Students have a solid understanding of the theoretical, the operational, and the implementation underpinnings of the modern computing infrastructure to be able to effectively utilize the whole spectrum of the modern computing infrastructure, including computer hardware, software, programming environments, operating systems, and networking environments. A successful student will be able to understand the basic components of a computer operating system, and the interactions among the various components. The students will also implement solutions via C/C++ programs.

Contents

1. Introduction: Over view of: Operating Systems, Operating-System Structure, Operating-System Operations, Process management, Memory Management, Storage Management, Protection and Security, Protection and Security, Distributed Systems, Special-Purpose Systems, Computing Environments.
2. Operating-System Structures: Operating-System Services, Operating-System Structure, User Operating-System Interface, Virtual Machines, System Calls, Operating System Generation, Types of System Calls, System Boot, System Programs.
3. Processes: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Communication in Client- Server Systems. Threads: Multithreading Models, Thread Libraries, Threading Issues.
4. CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Thread Scheduling, Algorithm Evaluation.
5. Process Synchronization: Background, Monitors, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization.
6. Deadlocks.
7. Main Memory: Swapping, Contiguous Memory Allocation, Paging.
8. Virtual Memory: Allocating Kernel Memory, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing.
9. File-System Implementation.
10. I/O Systems: STREAMS, Hardware, Performance, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O Requests to Bibliographical Notes, Hardware Operations.
11. Security.

Recommended Texts

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2014). *Operating system concepts essentials*. New York: John Wiley & Sons, Inc.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems*. New York: Pearson.
2. Stallings, W. (2009). *Operating systems: internals and design principles*. New York: Pearson/Prentice Hall.

CMPC-5102 **Software Engineering** **3(3+0)**

Software Engineering comprises the core principles consistent in software construction and maintenance: fundamental software processes and life-cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and standard notations, principles of software architecture and re-use, software quality frameworks and validation, software development, and maintenance environments and tools. An introduction to object-oriented software development process and design. Topics include: iterative development, interpretation of requirements and use case documents into code; application of design notation in UML and use of commonly-used design patterns. Current industry-strength programming languages, technologies and systems feature highly in the practical components, electives and projects of the course, but they are also taught with a view to understanding and applying principles underlying their more ephemeral character. This course introduces students to the tools and techniques for developing software systems. The course teaches the fundamental strategies of abstraction, decomposition and reuse as methods for constructing such systems. Verification and validation techniques, with an emphasis on testing, are taught as a means to ensure that students are able to deliver software products of the quality required.

Contents

1. The Nature of Software, Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths.
2. Generic Process Models
3. Specialized Process Models
4. Introduction to Systems Analysis and Design, Business Information Systems, Information System Components, Types of Information Systems, Evaluating Software, Make or Buy Decision.
5. Introduction to SDLC, SDLC Phases, System Planning, Preliminary Investigation, SWOT Analysis.
6. The Importance of Strategic Planning, Information Systems Projects, Evaluation of Systems Requests, Preliminary Investigation, Systems Analysis, Requirements Modeling, Fact-Finding Techniques.
7. Requirements Engineering.
8. Requirements Modelling Strategies, Difference between Structured Analysis and Object-Oriented Analysis; Difference between FDD Diagrams & UML Diagrams.
9. Data & Process Modelling, Diagrams
10. Design Within the Context of Software Engineering, The Design Process, Design Concepts, Design Models: Data Design Elements.
11. Architecture Design Elements, Interface Design Elements, Component-Level Design Elements, Deployments Design Elements.
12. System Architecture, Architectural Styles, User Interface Design:
13. Software Quality Assurance.
14. Introduction to Project Management.

Recommended Texts

1. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. New York: McGraw-Hil
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Sommerville, I. (2011). *Software engineering* (9th ed.). New York: Pearson Education, Inc.
2. Shelly, G. B., & Rosenblatt, H. J. (2011). *Systems analysis and design*. Boston: Cengage Learning.

CMPC-5205 **Computer Networks** **4(3+1)**

This course describes the key terminologies and technologies of computer networks. It explains the services and functions provided by each layer in the Internet protocol stack. It identifies various internetworking devices and protocols, and their functions in a network. Analyze working and performance of key technologies, algorithms and protocols. Build Computer Network on various Topologies. It considers how to design networks and protocols for diverse situations, analyses several application and support protocols from a distributed systems viewpoint, and identifies significant problem areas in networked communications. Upon successful completion, students will have the knowledge and skills to describe, analyse and evaluate various related technical, administrative and social aspects of specific computer network protocols from standards documents and other primary materials found through research, analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies. Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols.

Contents

1. Introduction to networks and protocols architecture
2. Basic concepts of networking, network topologies and the Internet
3. Layered architecture and the OSI model
4. Physical layer functionality, data link layer functionality and the TCP/IP protocol architecture
5. Multiple access techniques, WAN Technologies and protocols, circuit switching and packet switching.
6. Wireless networks, Cellular Network Generations and LTE-Advanced
7. LAN technologies, LAN protocol architecture and virtual LANs
8. MAC addressing.
9. Networking devices, bridges, hubs and switches
10. Network layer protocols, Principles of Internetworking, IPv4 and IPv6.
11. IP addressing, Internet Protocol Operation, virtual private networks and IP Security and Sub-netting, CIDR.
12. Transport layer protocols, ports and sockets and connection-oriented transport protocol mechanisms
13. Routing protocols OSPF, EIGRP, RIP and routing in packet-switching networks
14. Connection establishment, flow and congestion control, effects of congestion, TCP congestion control and datagram congestion control protocol
15. Application layer protocols, electronic mail (SMTP and MIME), Web Access: HTTP and DNS
16. Latest trends in computer networks, real-time traffic and voice over IP

Pre-Requisite: Nil

Recommended Texts

1. Stallings, W. (2007). *Data and computer communications*. New York: Pearson Education, Inc.

Suggested Readings

1. Kurose, J. F. (2005). *Computer networking: A top-down approach featuring the internet*. New Delhi: Pearson Education India.
2. Tannenbaum, A. S. (2002). *Computer networks*. New Delhi: Pearson Education India.

CMPC-6201 **Database Systems** **4(3+1)**

This course is intended to provide you with an understanding of the current theory and practice of database management systems. To help you more fully appreciate their nature, the course provides a solid technical overview of database management systems, using a current database product as a case study. In addition to technical concerns, more general issues are emphasized. These include data independence, integrity, security, recovery, performance, database design principles, and database administration. The course introduces students to the fundamental Database concepts and their application in the workplace. Objectives include basic understanding of Database systems and their working as well as use. They will also learn Design conceptual, logical and physical database schemas using different data models. They will use Structured Query Language (SQL) for database definition and manipulation in any DBMS. Upon successful completion of this course, students should be able to describe the fundamental elements of relational database management systems.

Contents

1. Databases Overview: Basics and fundamentals
2. Data Models, Advantages of DB, Cost and Risk, Components of the DB Environment.
3. E-R Model, Modelling Rules, Modelling Entities, Attributes and relations
4. Enhanced E-R Model, Subtype/Super-types, Specifying Constraints in Relationships
5. Relational Data Model, Integrity Constraints, Transforming EER Diagrams into Relations
6. Introduction to Normalization, first, second and third normal form. Dependencies and key
7. Designing Physical DB, Designing Fields: Data Types, Coding Techniques, Handling Missing Data
8. Deformalizing and Partitioning Data, File Organizations
9. Introduction to SQL, The SQL Environment, Defining a Database in SQL
10. Basic SQL Commands for Processing Single Tables
11. SQL Commands for Processing Multiple Tables using Joins and Subqueries
12. Client/Server Architectures, Databases in a Two-Tier and Three-Tier Architectures
13. Web Application Components, Databases in Three-Tier Applications
14. The Roles of Data and Database Administrators, Database Backup and Recovery, Types of Database Failure, Disaster Recovery
15. Concurrent Access, Serializability, Locking Mechanisms, Dictionaries & Repositories
16. Lab work should be carried out to develop students' Database Skills.

Recommended Texts

1. Hoffer, J. A., Ramesh, V., & Topi, H. (2011). *Modern database management*. New York: Prentice Hall,

Suggested Readings

1. Elmasri, R., & Navathe, S. (2017). *Fundamentals of database systems* (Vol. 7). New York: Pearson.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (1997). *Database system concepts* (Vol. 5). New

York: McGraw-Hill.

CMPC-6101 **Information Security** **3 (3+0)**

At the end of the course, the students will be able to explain key concepts of information security. They will be able to discuss legal, ethical, and professional issues in information security. They can apply various security and risk management tools for achieving information security and privacy. Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications. Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath. Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges. Determine appropriate mechanisms for protecting information systems ranging from operating systems to database management systems and to applications. It will be easy to identify appropriate techniques to tackle and solve problems in the discipline of information security.

Contents

1. Information security foundations, security design principles; security mechanisms.
2. Vulnerabilities and protections: Malwares.
3. Hash functions.
4. Digital signatures.
5. Key management.
6. Authentication: Access control.
7. Symmetric cryptography: Symmetric Encryption, DES,
8. AES, Cipher Block Modes.
9. Asymmetric cryptography: HMAC, The RSA, Diffie-Hellman and Other Algorithms Software security
10. Database security: The need, DBMS Relational Databases, SQL Injection Attacks, Database Access Control, Inference, Database Encryption
11. Network security: Secure E-Mail and S/MIME, Domain Keys Identified Mail, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, Wireless Security, Mobile Device Security
12. Firewalls: The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Firewall Basing.
13. Intrusion detection: Intruders, Intrusion Detection, Analysis Approaches, Types of IDS.
14. Security policies, policy formation and enforcement, risk assessment.
15. Cybercrime, law and ethics in information security, privacy and anonymity of data

Recommended Texts

1. Vermaat, M. E., Sebok, S. L., Freund, S. M., Campbell, J. T., & Frydenberg, M. (2017). *Discovering computers© 2018: digital technology, data, and devices*. Mason: Cengage Learning, Inc
2. Whitman, M. E., & Mattord, H. J. (2011). *Principles of information security*. Mason: Cengage Learning.

Suggested Readings

1. Easttom, W. (2013). *Computer security fundamentals with my IT certification lab bundle*. New York: Pearson IT Certification.

CMPC-6701 Capstone Project-I 3(0+3)

The Capstone Project is an opportunity for participants to put their leadership competencies into practice. Capstone Project allows the student to identify and develop a project that puts into practice the leadership skills and competencies learned during the courses of study. In Capstone Project I students are working in a group form and finalize the Project Proposal. On approval of Project proposal from Project Coordinator group has to work on finalizing the Software Requirement Specifications. In this phase groups are utilizing the Software Engineering guidelines to finalize the Software Requirement Specifications. On finalization of Requirement documentation software Design process is initiated. In this guideline of software design specifications are followed. By developing Project potters and videos groups can improve and sharpen their multimedia skills. Posters and Videos competitions are held regularly to improve competition among project groups. At the end each group has to give presentation and explain in detail all the work completed during semester.

Contents

1. Identifying Project Area
2. Filtering and Finalizing Project
3. Project Proposal
4. Identifying Project Scope
5. Finalizing project Description
6. Non-Functional Requirements
7. Functional Requirements
8. Identifying Project High level Plans
9. Overall project Description
10. System Architecture
11. Literature Review (In case of Research base project)
12. Software Cost, Time and Effort Estimations
13. Identifying main use cases.
14. Finalizing Software Development tool
15. Project Poster Development
16. Project Video Development

Pre-Requisite: Databases System, Software Engineering.

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction*. Switzerland: Springer International Publishing.
2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addison-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc Graw Hill
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York: Pearson.

CMPC-6702 Capstone Project-II 3(0+3)

Capstone Project II is a continuation of Capstone Project I. This course provides students with the opportunity to apply the knowledge and skills acquired in their courses to a specific problem or issue. To allow students to extend their academic experience into areas of personal interest, working with new ideas, issues, organizations, and individuals. Main focus of this course is development of Capstone Project I and according to details and area finalized. Group are developing functional requirements and collaborate with the project supervisors and coordinators throughout the semester. At the end of project groups has to present running project to the Project evaluation Committee. In Capstone Project II student can consult with the supervisor or expert from industry. Capstone Project II engages the students with real world market-oriented problems and complexities. Groups has to put in extra effort along with final semester courses to cover the complexities and deadlines of Capstone Project II.

Contents

1. System Sequence Diagram,
2. Domain Model
3. State Chart Diagrams and Implementation
4. Modeling Generalization, Design Class Diagram, Mapping Model to Domain Model
5. Implementation of Design Class Diagram, Coding patterns, Mapping Design to Code
6. Implementation according to Project Area, Web base application, Android/IOS Development, Research base project, Network base project development, Cloud based project, Hardware supported, IOT based, AI and Machine Learning based projects.
7. Implementation design according to project development area.
8. For Database oriented projects details in points 9-10 are followed.
9. SQL Commands for Processing Multiple Tables using Joins and Sub queries.
10. Client/Server Architectures, Databases in a Two-Tier Architecture, Three-Tier Architectures.
11. Web Application Components, Databases in Three-Tier Applications.
12. For Network/cloud based/AI and Machine Learning based projects implementation according to area specification.
13. Project framework development according to specific area.
14. Applying Software Quality Assurance.
15. Software Testing Strategies, Strategic Issues, Test.
16. Validation, System and Integration Testing.

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction*. Switzerland: Springer International Publishing.
2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addison-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc Graw Hill
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York: Pearson.

MATH-5101 Calculus and Analytical Geometry 3(3+0)

The course let students to understand the foundation and basic ground for calculus and analytical geometry background. This course also gives the basic of derivation, integration and geometry. The primary objective is to endow the knowledge of basic concepts of calculus and geometry. Purpose of this course is to build the student's knowledge of differential/integral calculus of multi-variable functions based on their past experience of differential/integral calculus and analytic geometry of functions of one independent variable. After the successful completion of course, the students will be able to learn to analyze and solve problems relating analytical geometry, vector analysis & vector calculus and initial value problems. Students should be able to use their knowledge of algebra and limits to derive differentiation formulas, properties of derivatives and integrals, and to gain a basic understanding of the Fundamental Theorem and other theorems of calculus. Students should be able to determine appropriate techniques and knowledge necessary to solve mathematical or applied problems involving calculus.

Contents

1. Real Numbers and the Real Line, Coordinates, Lines, and Increments, Functions, Shifting Graphs, Trigonometric Functions.
2. Limits and Continuity: Rates of Change and Limits, Rules for Finding Limits, Target Values and Formal Definitions of Limits.
3. Derivatives: The Derivative of a Function, Differentiation Rules, Rates of Change.
4. Applications of Derivatives: Extreme Values of Functions, The Mean Value Theorem, The First Derivative Test for Local Extreme Values.
5. Integration: Indefinite Integrals, Integration by Substitution—Running the Chain Rule Backward, Estimating with Finite Sums, Riemann Sums and Definite Integrals, Properties, Area, and the Mean Value Theorem. Substitution in Definite Integrals. Numerical Integration.
6. Applications of Integrals: Areas between Curves, Finding Volumes by Slicing, Volumes of Solids of Revolution—Disks and Washers. Cylindrical Shells. Lengths of Plan Curves, Areas of Surfaces of Revolution, Moments and Centers of Mass.
7. Transcendental Functions: Inverse Functions and Their Derivatives, Natural Logarithms, The Exponential Function, ax and logax, Growth and Decay, L'Hôpital's Rule, Relative Rates of Growth.
8. Conic Sections, Parameterized Curves, and Polar Coordinates: Conic Sections and Quadratic Equations. Classifying Conic Sections by Eccentricity. Quadratic Equations and Rotations.
9. Vectors and Analytic Geometry in Space, Vectors in the Plane Dot Products, Vector-Valued Function Cartesian (Rectangular) Coordinates and Vectors in Space. Dot Products. Cross Products. Lines and Planes in Space. Cylinders and Quadric Surfaces. Cylindrical and Spherical Coordinates.

Recommended Texts

1. Fraleigh, J. B., & Katz, V. (1985). *Calculus with analytic geometry*. Boston: Addison-Wesley.
2. Swokowski, E. W. (1979). *Calculus with analytic geometry*. Milton Park: Taylor & Francis.

Suggested Readings

1. Anton, H., Bivens, I. C., & Davis, S. (2009). *Calculus: early transcendentals*. Hoboken: John Wiley.
2. Anton, H., & Herr, A. (1988). *Calculus with analytic geometry*. New York: Wiley

MATH-5102 **Probability and Statistics** **3(3+0)**

The theory and methods of Statistics play an important role in all walks of life, society, medicine and industry. They enable important understanding to be gained and informed decisions to be made, about a population by examining only a small random sample of the members of that population. Probability theory and statistical distributions are needed to quantify this uncertainty, and assess the accuracy of our inference about the population. This module aims to lay foundations in probability and distribution theory, data analysis and the use of a statistical software package, which will be built upon in later modules. The concepts of random variables as outcomes of random experiments are introduced and the key properties of the commonly used standard univariate random variables are studied. Emphasis is placed on learning the theories by proving key properties of each distribution. At the end of the course the students will be able to understand the concepts of data analysis, presentation, counting techniques, probability and decision making.

Contents

1. Introduction to Statistics and Data Analysis, Role of Probability.
2. Discrete and Continuous Data, Statistical Modeling, Scientific Inspection, and Graphical, Types of Statistical Studies, Observational Study, and Retrospective Study.
3. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event.
4. Random Variables and Probability Distributions
5. Mean, Variance and Covariance of Random Variables, Chebyshev‘s Theorem.
6. Binomial and Multinomial Distributions, Hypergeometric Distribution, Negative Binomial and Geometric Distributions, Poisson Distribution and the Poisson Process.
7. Continuous Probability Distributions.
8. Fundamental Sampling Distributions and Data Descriptions.
9. One- and Two-Sample Estimation Problems: Statistical Inference, Classical Methods of Single Sample.
10. Estimating a Proportion, Estimating the Difference between Two Proportions, Single Sample: Estimating the Variance, Estimating the Ratio of Two Variances.
11. One- and Two-Sample Tests of Hypotheses: General Concepts, testing a Statistical Hypothesis, The Use of P-Values for Decision Making in Testing Hypotheses.
12. Single Sample.
13. One- and Two-Sample Tests Concerning Variances, Goodness-of-Fit Test, Test for Independence and Homogeneity
14. Simple Linear Regression and Correlation: The Simple Linear Regression Model, Least Squares and the Fitted Model, Properties of the Least Squares Estimators.
15. Multiple Linear Regression and Certain

Recommended Texts

1. Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (1993). *Probability and statistics for engineers and scientists* (Vol. 5). New York: Macmillan.

Suggested Readings

1. Schiller, J., Srinivasan, R., & Spiegel, M. (2012). *Probability and statistics*. New York: McGraw-Hill Education.
2. Haigh, J. (2012). *Probability: A very short introduction*. Oxford: Oxford University Press.

MATH-5103 Linear Algebra 3(3+0)

In this course we study a branch of mathematics, called linear algebra, and some of its applications. Students studying this course should master properties of matrices including how to use them to solve linear systems of equations and how they are used in linear transformations between vector spaces. Students are also expected to gain an appreciation for the applications of linear algebra to areas such as computer science, engineering, biology and economics. Upon successful completion of this course students will be able to use computational techniques and algebraic skills essential for the study of systems of linear equations, matrix algebra, vector spaces, eigenvalues and eigenvectors. Find the dimension of spaces such as those associated with matrices and linear transformations. Students will learn to prove basic results in linear algebra using appropriate proof-writing techniques such as linear independence of vectors, properties of subspaces, linearity, injectivity and surjectivity of functions.

Contents

1. Introduction to Vectors: Vectors and Linear Combinations, Lengths and Dot Products, Matrices.
2. Solving Linear Equations: Vectors and Linear Equations, the Idea of Elimination, Elimination Using Matrices, Rules for Matrix Operations, Inverse Matrices.
3. Elimination = Factorization; A = LU, Transposes and Permutations
4. Vector Spaces and Subspaces: Spaces of Vectors, The Null space of A: Solving Ax = 0,
5. The Rank and the Row Reduced Form, the Complete Solution to Ax = B, Independence, Basis and Dimension, Dimensions of the Four Subspaces.
6. Orthogonally: Orthogonally of the Four Subspaces, Projections, Least Squares Approximations, Orthogonal Bases and Gram-Schmidt.
7. Determinants: The Properties of Determinants, Permutations and Cofactors, Cramer’s Rule, Inverses, and Volumes.
8. Eigenvalues and Eigenvectors: Introduction to Eigenvalues, Diagonalizing a Matrix, Applications to Differential Equations, Symmetric Matrices, Positive Definite Matrices, Similar Matrices, Singular Value Decomposition (SVD).
9. Applications: Matrices in Engineering, Graphs and Networks, Markov Matrices, Population, and Economics; Linear Programming, Fourier series: Linear Algebra for Functions, Linear Algebra for Statistics and Probability, Computer Graphics.
   Numerical Linear Algebra: Gaussian Elimination in Practice, Norms and Condition
10. Numbers, Iterative Methods for Linear Algebra.
11. Complex Vectors and Matrices: Complex Numbers, Hermitian and Unitary Matrices, Matrix Factorizations.

Recommended Texts

1. Strang, G., Strang, G., Strang, G., & Strang, G. (1993). *Introduction to linear algebra* (Vol. 3). Wellesley, MA: Wellesley-Cambridge Press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Kreyszig, E. (2011). *Advanced engineering mathematics*. New York: John Wiley & Sons Inc.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

PHYS-5101 Applied Physics 3(3+0)

Practical research experience is a valuable supplement to the formal course of instruction. Applied physics students participate in an informal undergraduate seminar to study current and practical problems in applied physics, and obtain hands-on experience in at least two advanced laboratory courses. Objectives of the course includes understanding of the fundamental concepts/laws in physics by explaining and discussing the physics as well as their relevance to everyday events and circumstances in a broad interdisciplinary context. Reveal critical thinking/ analytical reasoning ability by setting up mathematical descriptions of physical systems and to calculate measurable quantities that provide an understanding of the physical environment in terms of the concepts listed in the course content. Describe the major concepts in physics. Demonstrate an appropriate level of competency in both computer and research laboratory skills. Communicate in written and oral forms with interested citizens and professionals on key concepts in physics and general scientific issues.

Contents

1. Zero Reference Level, Chassis Ground, Ohm ‘s Law, Formula Variations of Ohm ‘s Law
2. Graphical Representation of Ohm ‘s Law, Linear Resistor, Non-Linear Resistor
3. Cells in Series and Parallel.
4. Resistive Circuits.
5. Resistors (5.1-15)
6. Inductors (5.19-21)
7. Capacitors (5.35-48).
8. Energy Sources. Magnetism and electromagnetism Solid State.
9. Atomic structure, Electron distribution of different atoms, Energy bands in solids, Bonds in solids, Conduction in solids
10. Conductors, Semiconductors and types of semiconductors
11. Insulators, Majority and Minority charge carriers
12. Mobile charge carriers and immobile ions Drift current in good conductors.
13. Modulation and Demodulation. Carrier Waves, Modulation, Demodulation or Detection, Comparison between Amplitude Modulation (AM) and Frequency Modulation (FM).
14. Integrated Circuits. Advantages of ICs and Drawbacks of ICs, Scale of Integration, Classification of ICs by function, Linear and Digital Integrated Circuits, IC Terminology, Fabrication of IC Components, Popular Application of ICs, Operational Amplifier.
15. Fibre Optics. Structure of Optical Fibres, Classification of Optical Fibres, Fibre Characteristics, Choice of Wavelength, Optical Fibre cable, Application of Fibre Optic Communication.

Pre-Requisite: Nil

Recommended Texts

1. Theraja, B. L. (2005). *Basic electronics: solid state*. New Delhi: S. Chand Limited.

Suggested Readings

1. Paul, A. (1999). *Electronic principles*. New York: McGraw-Hill.

ENGL-5101 English Composition and Comprehension 3(3+0)

In this course, students will learn how to articulate clearly, take and pass on messages, deal with customers effectively, read, understand and follow a wide range of documents and write fluently and accurately, using accepted business conventions of format, spelling, grammar and punctuation. They will have strengthened student's skills which enable them to deal with the practical problems and challenges of life – at home, in education and at work. This course focuses on critical reading and strategies for varying writing style, tone, and form for multiple purposes and audiences. In this course students will learn to compose texts, including academic essays that clearly assert a claim and support the claim with compelling evidence. Besides, basics of grammar like parts of speech, tenses, vocabulary exercises are so designed to teach the students basic tools of English language and upgrade their ability to functional utilization of the language through the practical application of grammar rules.

Contents

1. Punctuation: Periods, Question Marks, Exclamation Marks, Semicolons, Colons etc.
2. Writing Mechanics: Capitals, Abbreviations; Vocabulary: Frequently Confused Words.
3. Vocabulary: Phrases, Synonyms, Antonyms, Idioms, General Vocabulary.
4. Use of Articles and One, A Little/ A Few, This, That; Interrogatives: Why-? Words and How?
5. Kinds of Nouns; Kinds of Adjectives; Adverbs: Kinds, Form, Position and Use.
6. Prepositions; Possessive, Personal and Reflexive Pronouns; Relative Pronouns and Clauses.
7. Classes of Verbs: Ordinary Verbs, Auxiliary Verbs etc.
8. The Present Tenses
9. The Past and Perfect Tenses
10. The Future; The Infinitive
11. The Gerund & the Participles; Commands, Requests, Invitations, Advice, Suggestions.
12. The Passive Voice; Indirect Speech
13. Conjunctions, Purpose
14. Clauses, Result, Concession, Comparison, Time, Numerals, Dates, and Weights and Measures; Spelling Rules; Phrasal Verbs etc.

Pre-Requisite: Nil

Recommended Texts

1. Wren, P. C., & Martin, H. (1999). *High school english grammar and composition*. New Delhi: S. Chand.

Suggested Readings

1. Thomson, A. J., & Martinet, A. V. (1980). *A practical english grammar*. Oxford: Oxford University Press.
2. Swan, M. (1984). *Basic english usage*. Oxford: Oxford University Press.

ENGL -5102 **Communication and Presentation Skill** **3(3+0)**

The aim is to teach the students to sensitize students to their communicative behavior and enable them to reflect and improve on their communicative behavior/performance. This course will enable you to build capacities for self-criticism and facilitate growth and it will lead students to effective performances in communication. The objective of this course is to create awareness of one's current presentation skills and practice & develop effective skills for presentation, understand the importance of communication types, developing effective verbal communication, Importance of grooming and etiquette and cross-cultural communication. Demonstrate improved interpersonal skills by identifying and developing a repertoire of strategies for improved communication effectiveness and demonstrate the strategies in oral and written contexts. Demonstrate improved research, organizational, and critical thinking skills by finding and evaluating reference material and organizing and presenting effective messages adapted to specific audiences. Outcomes will be assessed by a series of embedded class assessments, exams, and problem-solving exercises, by class assignments, presentations and projects, research assignments and self-analysis in oral and written contexts.

Contents

1. Communication Skills
2. Internal Representation
3. Elements of Communication
4. Listening: Real Vs. Introduction
5. Expressing
6. Clarifying Language
7. Making Contact
8. Prejudgment: Prejudgment Traps, Stereotypes
9. Influencing Others
10. Public Speaking
11. Preparing A Formal Oral Presentation
12. Delivering Presentation
13. Interviewing
14. Effective Written Communication
15. Building Rapport

Pre-Requisite: English Composition & Comprehension (English-I)

Recommended Texts

1. Training, M. T. D. (2012). *Effective communication skills*. Copenhagen: Bookboon.

Suggested Readings

1. McKay, M., Davis, M., & Fanning, P. (2009). *Messages: The communication skills book*. Oakland: New Harbinger Publications.
2. Bradbury, A. (2006). *Successful presentation skills* (Vol. 51) London: Kogan Page Publishers.

ENGL -6101 Technical and Business Writing 3(3+0)

The aim is to teach the students effectively plan and structure technical reports and to recognize the various stages in writing a technical report. The course focuses on the practice and study of selected types of discourse employed in professional business situations and helps prepare students for different kinds of writing they will encounter in their professional lives. By the end of this course, all students should be able to analyze and evaluate audience/purpose/situation as they apply to business writing contexts. Produce clear, concise, effective audience and purpose-specific business rhetoric. Incorporate process (research, invention, writing, revision, and editing) into all writing tasks and, through multiple drafts, create document fluency. Develop strategies to facilitate communication across ethnic and/or business cultures. Practice the unique qualities of professional rhetoric and writing style, such as sentence conciseness, clarity, accuracy, honesty, avoiding wordiness or ambiguity, using direct order organization, readability, coherence and transitional devices.

Contents

1. Writing for Readers
2. Discovering and Planning
3. Purpose, Thesis, and Audience
4. Drafting, Revising, Editing, and Proofreading
5. Paragraphs: Unfocused Paragraphs; Revising for Focus
6. Clear and Emphatic Sentences
7. Reasoning Critically
8. Arguing Persuasively
9. Designing Documents
10. Writing in Online Communities
11. Speaking Effectively
12. Academic Writing
13. Abstract, Informative Report
14. Public Writing
15. Researching and Writing

Pre-Requisite: Communication and Presentation Skills (English-II)

Recommended Texts

1. Anson, C. M., Schwegler, R. A., & Muth, M. F. (2005). *The longman writer's companion*. New York: Longman Publishing Group.

Suggested Readings

1. Pickett, N. A., & Laster, A. A. (1996). *Technical english: writing, reading, and speaking*. New York: Harpercollins College Division.
2. Brusaw, C. T., Alred, G. J., & Oliu, W. E. (1996). *The business writer's companion*. New York: St. Martin's Press.

URCI-5105 **Islamic Studies** **2(2+0)**

Islamic Studies engages in the study of Islam as a textual tradition inscribed in the fundamental sources of Islam; Qur'an and Hadith, history and particular cultural contexts. The area seeks to provide an introduction to and a specialization in Islam through a large variety of expressions (literary, poetic, social, and political) and through a variety of methods (literary criticism, hermeneutics, history, sociology, and anthropology). It offers opportunities to get fully introductory foundational bases of Islam in fields that include Qur'anic studies, Hadith and Seerah of Prophet Muhammad (PBUH), Islamic philosophy, and Islamic law, culture and theology through the textual study of Qur'an and Sunnah. Islamic Studies is the academic study of Islam and Islamic culture. It majorly comprises of the importance of life and that after death. It is one of the best systems of education, which makes an ethical groomed person with the qualities which he/she should have as a human being. The basic sources of the Islamic Studies are the Holy Qur'an and Sunnah or Hadith of the Holy Prophet Muhammadﷺ. The learning of the Qur'an and Sunnah guides the Muslims to live peacefully.

Contents

1. Study of the Qur'an (Introduction to the Qur'an, Selected verses from *Surah Al-Baqarah, Al-Furqan, Al-Ahzab, Al-Mu'minoon, Al-An'am, Al-Hujurat, Al-Saff*)
2. Study of the Hadith (Introduction to Hadith literature, Selected Ahadith (Text and Translation)
3. Introduction to Qur'anic Studies
4. Basic Concepts of Qur'an
5. History of Quran
6. Basic Concepts of Hadith
7. History of Hadith
8. Kinds of Hadith
9. Uloom –ul-Hadith
10. Sunnah & Hadith
11. Seerat ul-Nabi (PBUH), necessity and importance of Seerat, role of Seerah in the development of personality, Pact of Madinah, Khutbah Hajjat al-Wada' and ethical teachings of Prophet (PBUH).
12. Legal Position of Sunnah
13. Islamic Culture & Civilization
14. Characteristics of Islamic Culture & Civilization
15. Historical Development of Islamic Culture & Civilization
16. Comparative Religions and Contemporary Issues
17. Impact of Islamic civilization

Recommend Texts

1. Hassan, A. (1990). *Principles of islamic jurisprudence*. New Dehli: Adam Publishers.
2. Zia-ul-Haq, M. (2001). *Introduction to al-Sharia al-islamia*. Lahore: Aziz Publication.

Suggested Readings

1. Hameedullah, M. (1957). *Introduction to islam*. Lahore: Sh M Ashraf Publisher.
2. Hameedullah, M. (1980). *Emergence of islam*. New Dehli: Adam Publishers.
3. Hameedullah, M. (1942). *Muslim conduct of state.* Lahore: Sh M Ashraf Publisher.

URCP-5106 Pakistan Studies 2(2+0)

The course is designed to acquaint the students of BS Programs with the rationale of the creation of Pakistan. The students would be apprised of the emergence, growth and development of Muslim nationalism in South Asia and the struggle for freedom, which eventually led to the establishment of Pakistan. While highlighting the main objectives of national life, the course explains further the socio-economic, political and cultural aspects of Pakistan's endeavors to develop and progress in the contemporary world. For this purpose, the foreign policy objectives and Pakistan's foreign relations with neighboring and other countries are also included. This curriculum has been developed to help students analyse the socio-political problems of Pakistan while highlighting various phases of its history before and after the partition and to develop a vision in them to become knowledgeable citizens of their homeland. Have an in-depth understanding about the working and impacts of a few major civil engineering projects in Pakistan.

Contents

1. Contextualizing Pakistan Studies
2. Geography of Pakistan: Geo-Strategic Importance of Pakistan
3. Freedom Movement (1857-1947)
4. Pakistan Movement (1940-47)
5. Muslim Nationalism in South Asia
6. Two Nations Theory
7. Ideology of Pakistan
8. Initial Problems of Pakistan
9. Political and Constitutional Developments in Pakistan
10. Economy of Pakistan: Problems and Prospects
11. Society and Culture of Pakistan
12. Foreign Policy Objectives of Pakistan and Diplomatic Relations
13. Current and Contemporary Issues of Pakistan
14. Human Rights: Issues of Human Rights in Pakistan

Recommended Texts

1. Kazimi, M. R. (2007). *Pakistan studies.* Karachi: Oxford University Press.
2. Sheikh, Javed Ahmad (2004). *Pakistan's political economic and diplomatic dynamics.* Lahore: Kitabistan Paper Products.

Suggested Readings

1. Hayat, Sikandar (2016). *Aspects of pakistan movement.* Islamabad: National Institute of Historical and Cultural Research.
2. Kazimi, M. R (2009). *A concise history of pakistan.* Karachi: Oxford University Press.
3. Talbot, Ian (1998). *Pakistan: a modern history*. London: Hurst and Company.

ICTC-5201 Introduction to Information & Communication Technologies 3(2+1)

The course introduces students to information and communication technologies and their application in the workplace. Objectives include basic understanding of computer software, hardware, and associated technologies. How computers can be used in the workplace, how communications systems can help boost productivity, and how the Internet technologies can influence the workplace. The main objective of the course is to make the students familiar to the basic knowledge of information and communication technologies. Students will be able to independently use information technology tools during the course of the study and, subsequently for searching information and preparing reports, project assignments, presentations, and other materials. The objective is also to present trends in ICT and to prepare students to follow them. They will also learn Open office being used on other operating systems and platforms. Specific software's related to specialization areas are also part of course. Course will also cover Computer Ethics and related Social media norms and cyber laws.

Contents

1. Basic Definitions & Concepts
2. Hardware: Computer Systems & Components, Storage Devices, Number Systems
3. Software: Operating Systems, Programming and Application Software,
4. Introduction to Programming
5. Databases and Information Systems Networks
6. Data Communication
7. The Internet, Browsers and Search Engines
8. Email Collaborative Computing and Social Networking
9. E-Commerce
10. IT Security and other issues

Pre-Requisite: Nil

Recommended Texts

1. Vermaat, M. E., Sebok, S. L., Freund, S. M., Campbell, J. T., & Frydenberg, M. (2017). *Discovering computers© 2018: digital technology, data, and devices*. Mason, OH: Nelson Education.

Suggested Readings

1. O'Leary, T., O'Leary, L., & O'Leary, D. (2016). *Computing essentials 2017*. San Francisco: McGraw-Hill Education Group.
2. Fuller, F. (2008). *Computers: understanding technology*. Boulder: Paradigm Publishing, Incorporated.

ITSC-5101 **Professional Practices** **3(3+0)**

This course is a graduate-level course of Professional Practices. The course aims to elaborate foundation of Professional Practices as a subject. It focuses on both mainstream and critical approaches to visualize and examining how these topics conceptualize Professional Practices as a field of study. The course explicitly relates Professional Practices as cognate. Students will gain confidence in their ability to communicate by practicing and receiving feedback on business communication skills. Through this critical enquiry process, it is expected that students will further develop their understandings of their own practices and change and develop aspects of these practices. It has a particular emphasis on building professional knowledge of working in a business environment and developing work ready skills in the areas of written and oral communication, intercultural communication, client service, problem solving and self-management. Students will develop strategies to communicate these abilities through career communication, including career documents and interviews. Opportunities for improving academic and workplace language proficiency are embedded in the course.

Contents

1. The Engineering Profession.
2. The Structure of Organizations.
3. Finance and Accounting.
4. Anatomy of a Software House.
5. Computer Contracts.
6. Intellectual Property Rights.
7. The Framework of Employee Relations Law and Changing Management Practices.
8. Human Resource Management and Software Engineering.
9. Health and Safety at Work.
10. Software Liability: Liability and Practice.
11. Computer Misuse and the Criminal Law.
12. Graphical Methods for Comparing Means.
13. Regulation and Control of Personal Information: Data Protection, Defamation and Related Issues.
14. The British Computer Society Code of Conduct.

Pre-Requisite: Nil

Recommended Texts

1. Bott, F., Coleman, A., & Rowland, D. (2000). *Professional issues in software engineering*. Florida: CRC Press.

Suggested Readings

1. Sara Baase. *A gift of fire: social, legal, and ethical issues for computing and the internet* (3^{rd} ed.). Harlow: Prentice Hall

BUSB-5101 **Business Economics** **3(3+0)**

Business Economics, also called Managerial Economics, is the application of economic theory and methodology to business. Business involves decision-making. Decision making means the process of selecting one out of two or more alternative courses of action. Business economic meets these needs of the business firm and provides an analysis of the behavior of private firms operating in different types of market structure. Economics is not a collection of facts to be written down and memorized. Economics is a way of thinking about the world and the world is always changing. Economists have developed a set of simple but widely applicable concepts and principles that are useful for understanding economic situations ranging from decisions that individuals make every day to decisions made by firms and governments in complex markets. The objective of this course is to help students learn and understand these concepts and principles and to apply them to a variety of economic situations.

Contents

1. Introduction to Business.
2. Introduction to economic environment.
3. The working of competitive markets.
4. Demand and the consumer.
5. Supply decisions in a perfectly competitive market.
6. Pricing and output decisions in imperfectly competitive markets.
7. Business growth and strategy.
8. Multinational corporations and business strategy in a global economy.
9. Government, the firm and the market.
10. The economy and business activity.
11. National macroeconomic policy.
12. The global trading environment.

Pre-Requisite: Nil

Recommended Texts

1. Sloman, J., Garratt, D., Guest, J., & Jones, E. (2016). *Economics for business*. Harlow: Pearson Education.

Suggested Readings

1. Griffiths, A., & Wall, S. (2008). *Economics for business and management*. Harlow: Pearson Education.
2. McAleese, D. (2004). *Economics for business: competition, macro-stability, and globalization*. Edinburgh Gate: Pearson Education.

BUSB-5107 Human Resource Management 3(3+0)

Human resource plays an important role in success and failure of any organization. The quality of human resource deployed by any organization determines the future of that organization. In the modern world the corporate sector puts lot of effort and resources on hiring and training their human resources to increase the working efficiency of their employees. The principal objective of this course is to prepare students to gain knowledge in the field of human resource management and enabling them to understand the human resource functions and practices in banks/organization for improved performance and help create a transparent organizational culture. The course also provides an overall understanding of organizational behavior concepts to assist students in recognizing organizational structure, culture and development concepts so that they are better equipped to perform in the organization, make informed decisions and effectively manage supervisors and subordinates for enhanced performance. A section on business ethics is also included to ensure conceptual understanding, need and application of these concepts in daily business transactions.

Contents

1. Introduction:
2. Concepts of human resource management
3. Human Resource Challenges
4. Human Resource Functions,
5. Philosophical approaches to Human Resource Management Job Design and Analysis:
6. An overview of Job design,
7. Techniques of job design,
8. Job analysis, Collection of job information, Applications of job analysis information Human Resources.
9. Planning & Recruitment:
10. Significance of Human Resource Planning, The planning process, the implementation of program,
11. Recruitment & selection policy issues, Source of recruitment, Selection process & procedure,
12. Evaluation of Human resource Planning &Recruitment, Career
13. Planning & Development: Promotion, Anachronism, Demotion,
14. Separation.
15. Training and Development:
16. Significance of training & development, Principles of training & development,
17. Training & development methods, Evaluation of

Recommended Books

1. Werther Jr, W. B., & Davis, K. (1985). *Personnel management and human resources*. New York: McGraw-Hill

Recommended Readings

1. Bernardin & Russell: *Human resource management*. New York: McGraw Hill
2. Cameron, K. S. and Quinn, R. E. (2011) *Diagnosing and changing organizational culture: based on the competing values framework.* (3rd ed.). San Francisco: Jossey-Bass

BUSB-5102 Principles of Management 3(3+0)

Formally defined, the principles of management are the activities that "plan, organize, and control the operations of the basic elements of [people], materials, machines, methods, money and markets, providing direction and coordination, and giving leadership to human efforts, so as to achieve the sought objectives of the enterprise. At the most fundamental level, management is a discipline that consists of a set of five general functions: planning, organizing, staffing, leading and controlling. These five functions are part of a body of practices and theories on how to be a successful manager. Managers are required in all the activities of organizations: budgeting, designing, selling, creating, financing, accounting, and artistic presentation; the larger the organization, the more managers are needed. Everyone employed in an organization is affected by management principles, processes, policies, and practices as they are either a manager or a subordinate to a manager, and usually they are both. The course will cover topics fundamentals and principles of management, administrative policy, objectives, and procedures and problems of organizational control and leadership.

Contents

1. Introduction to Managers and Management: What as Management and What Do Managers Do? Defining Management, Management Functions.
2. Management Roles, Management Skills.
3. Organizational Culture and Environment: The Manager: Omnipotent or Symbolic? The Organization's Culture.
4. The Environment - Defining Environment, the Specific Environment.
5. Decision Making the Essence of Manager 's Job: The Decision-Making Process, The Rational Decision Maker,
6. Decision Making Styles, Analysing Decision Alternatives – Certainty, Risk.
7. Planning: The Foundations of Planning, The Definition of Planning, Purposes of Planning, Types of Plans.
8. Contingency Factors on Planning, Objectives: The Foundation for Planning.
9. Organization Structure and Design: Defining Organization Structure and Design, Building, the Vertical Dimension of Organizations, Building the Horizontal Dimension of Organizations, the Contingency Approach to Organization Design, Application of Organization Design.
10. Motivation: Motivating Employees, what is Motivation? Contemporary Approaches to Motivation, Contemporary Issues in Motivation, From Theory to Practice.
11. Suggestions for Motivating Employees.

Recommended Texts

1. Robbins, S.P. & Coulter, Mary. (2008). *Management* (10^{th} ed.). New York: Mary, Prentice Hall.
2. Robbins, S.P. & DeCenzo, David A. (2010). *Fundamentals of management* (7^{th} ed.). New York: Prentice Hall.

Suggested Readings

1. Charles W. L. Hill and McShane S. (2006). *Principles of management* (1^{st} ed.). New York: McGraw-Hill/Irwin.
2. Carpenter M. (2009). *Principles of management*. Pitman: Flat World Knowledge.

BUSB 6124 Entrepreneurship and SME Management 3(3-0)

This course aims at helping students develop a positive, constructive and practical approach to: Entrepreneurship venture opportunities, innovations, change, fantasies, Environment of small business, sources and resolutions. Risk failure and new venture unit, feasibility of planning and concepts of planning, stages of growth model. The entrepreneur is defined as someone who has the ability and desire to establish, administer, and succeed in a startup venture along with risk entitled to it, to make profits. The best example of entrepreneurship is the starting of a new business venture. The entrepreneurs are often known as a source of new ideas or innovator, and bring new ideas in the market by replacing old with a new invention. This course is designed to help graduate in understanding the importance of new ventures at small scale. Important is to understand the difference between a usual business venture at small scale and entrepreneurship which is need of the modern times.

Contents

1. Introduction: entrepreneurial perspective,
2. Economics and entrepreneurship,
3. Process, ventures, practices and characteristics.
4. Entrepreneurship and new free enterprise: venture opportunities, innovations, Change, fantasies, environment of small business,
5. Sources and resolutions, corporate entrepreneurship, new venture unit of planning and concepts of planning,
6. Stages of growth model, responsibility of feasibility plan.
7. Product and services concepts and commercial opportunities (macro over view),
8. Products and technology, identification opportunities
9. Product development life cycle, product protection,
10. Trademark and patents, process of patents,
11. Validity of property rights and accessing government information
12. Human resources side of enterprise,
13. Infrastructure of services, types of service venture, success factors.
14. Marketing and new venture development, Marketing research for new ventures,
15. Marketing concepts, startup of marketing research, Market focused on organization, sources of market intelligence,
16. Competitive analysis and implications of market research
17. Marketing strategies and functions, Product concepts, 4 ps,
18. Entrepreneurial team and business formation

Recommended Texts

1. Holt, D. H. (1992). *Entrepreneurship.* Boston: Prentice Hall

Suggested Readings

1. Bolton, B. K., & Thompson, J. (2004). *Entrepreneurs: talent, temperament, technique*. Abingdon: Rutledge.

URCC-5110 Citizenship Education and Community Engagement 3(1+2)

In recent years, community engagement has become a central dimension of governance as well as policy development and service delivery. However, efforts to directly involve citizens in policy processes have been bedeviled by crude understandings of the issues involved, and by poor selection of techniques for engaging citizens. This course will provide a critical interrogation of the central conceptual issues as well as an examination of how to design a program of effective community engagement. This course begins by asking: Why involve citizens in planning and policymaking? This leads to an examination of the politics of planning, conceptualizations of "community" and, to the tension between local and professional knowledge in policy making. This course will also analyze different types of citizen engagement and examine how to design a program of public participation for policy making. Approaches to evaluating community engagement programs will also be a component of the course. Moreover, in order to secure the future of a society, citizens must train younger generations in civic engagement and participation. Citizenship education is education that provides the background knowledge necessary to create an ongoing stream of new citizens participating and engaging with the creation of a civilized society.

Contents

1. Introduction to Citizenship Education and Community Engagement: Orientation
2. Introduction to Active Citizenship: Overview of the ideas, Concepts, Philosophy and Skills
3. Identity, Culture and Social Harmony: Concepts and Development of Identity
4. Components of Culture and Social Harmony, Cultural & Religious Diversity
5. Multi-cultural society and inter-cultural dialogue: bridging the differences, promoting harmony
6. Significance of diversity and its impact, Importance and domains of inter-cultural harmony
7. Active Citizen: Locally active, globally connected
8. Importance of active citizenship at national and global level
9. Understanding community, Identification of resources (human, natural and others)
10. Human rights, Constitutionalism and citizens' responsibilities: Introduction to human rights
11. Universalism vs relativism, Human rights in constitution of Pakistan
12. Public duties and responsibilities
13. Social Issues in Pakistan: Introduction to the concept of social problem, Causes and solutions
14. Social Issues in Pakistan (Poverty, Equal and Equitable access of resources, unemployment)
15. Social Issues in Pakistan (Agricultural problems, terrorism & militancy, governance issues)
16. Social action and project: Introduction and planning of social action project
17. Identification of problem, Ethical considerations related to project
18. Assessment of existing resources

Recommended Texts

1. Kennedy, J. K. Brunold, A. (2016). *Regional context and citizenship education in asia and europe*. New York: Routledge Falmer.
2. Macionis, J. J. Gerber, M. L. (2010). *Sociology*. New York: Pearson Education

Suggested Books

1 British, Council. (2017). *Active citizen's social action projects guide.* Edinburgh: British Council

2 Larsen, K. A. (2013). *Participation in community work: international perspectives.* Vishanthic

Sewpaul, Grete Oline Hole.

ITCC-5101 Information Technology project Management 3(3+0)

Information Technology project management (ITPM) is the process of managing the plan, organization, and accountability to achieve information technology goals. Information Technology project managers' deal with the challenges of interdependent integrations, rapid technology upgrades, and version changes that can occur throughout the project timeline. Information technology projects are run by people. These people need to be selected wisely, they need to have the right mix of education, certification, training, and work experience. The course introduces students to information required to the process of software designing, maximizing the return from each stage of the software development life cycle. The course will introduce students about basic project management tools and their usage during SDLC. Information Technology Project management is the course that will help students in practice of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time.

Contents

1. Introduction: Software Project versus Other Type of Projects Dimensions of a Software Project, Project Vs. Program Management.
2. Introduction of PM Tools, PMI's Knowledge Areas, Technical Fundamentals in SPM, Lifecycle Relationships.
3. PMI Framework, PMI Process Groups: Process Initiating Process Group, Planning Process etc.
4. Understanding Organizations, Organizational Structures, Functional -Project –Matrix, identifying stakeholders: Define Responsibilities, Authority etc.
5. Project Planning: Project Selection, Project Scope, Project Infrastructure etc.
6. Project Evaluation: Strategic Assessment, Technical Assessment, Economic Assessment, Project Portfolio Management.
7. Selection of an Appropriate Approach in Project, Agile Process Model etc.
8. Software Effort Estimation: Work Breakdown Structure (WBS), Function Point Estimation, COCOMO and LOC.
9. Activity Planning: Project and Activities, Sequencings and Scheduling Activities
10. Identifying the Critical Path, Identifying the Critical Activities Project, AOA.
11. Risk Management: Categories of Risks; Evaluating the Risks to the Schedule: PERT.
12. Risk Control, RMMM, Configuration Control vs. Version Control.
13. Resource Allocation: Nature of Resources, Identifying Resource Requirements.
14. Monitoring & Control: Creating Framework, Collecting Data, Visualizing Progress.
15. Review and Evaluation: Determining Satisfaction of Requirements; Project closure.
16. Challenges of Outsourcing in Project Management, Presentations.

Recommended Texts

1. Hughes, B., & Cotterell, M. (2009). *Software project management*. New Delhi: Tata McGraw-Hill Education.
2. Rose, K. H. (2013). *A Guide to the project management body of knowledge* Newtown Square, Pa: Project Management Institute.

Suggested Readings

1. McConnell, S. (1998). *Software project survival guide*. Redmond: Pearson Education.
2. Wysocki, R. K. (2011). *Effective project management: traditional, agile, extreme*. New Jersey: John Wiley & Sons.

ITCC-6202 **System and Network Administration** **4(3+1)**

A network administrator is responsible for installing, maintaining and upgrading any software or hardware required to efficiently run a computer network. The IT or computer network may extend to a local area network, wide area network, the Internet and intranets. Duties of system administrators are usually charged with installing, supporting, and maintaining servers or other computer systems, and planning for and responding to service outages and other problems. Other duties may include scripting or light programming, project management for systems-related projects. Computer Management is an administrative tool included with Windows. The Computer Management console contains numerous standalone tools and utilities, including Task Scheduler, Device Manager, Disk Management and Services, that can be used to modify Windows settings and performance. This course will give an overview of systems and network administration based on both Windows and Linux environments. The objective are common system administration tasks and practices and how to implement and maintain standard services like email, file sharing, DNS and similar.

Contents

1. Introduction to System Administration, Systems, SA Components, building a Site from Scratch, growing a Small Site, Going Global, Replacing Services, moving a Data Centre, handling a High Rate of Office Moves, Assessing a Site
2. Server Environment (Microsoft and Linux): Known for Reliable Products, understand the Cost of Server Hardware, Consider Maintenance Contracts and Spare Parts, maintaining Data Integrity, Put Servers in the Data Centre, Client Server OS Configuration, and Provide Remote Console Access.).
3. Services and Comparative Analysis of most demanded OS: Important Attributes, Key Features, pros and Cons.
4. Linux Installation and verification
5. Configuring Local services and managing basic system issues.
6. Administer users and groups
7. Software Management.
8. Managing Network Services and Network monitoring tools.
9. Boot Management and Process Management.
10. IP Tables and filtering.
11. Securing network traffic.
12. Advance File systems and logs.
13. Bash Shell Scripting and Command line.
14. Configuring Servers (FTP, DNS, Apache) LAB.
15. Configuring Servers Cont. (DHCP, Samba, NFS) LAB.
16. Configuring Active Directory on Windows Server 2012 LAB

Pre-Requisite: Nil

Recommended Texts

1. Sanghera, P. (2011). *RFID+ Study guide and practice Exams: study guide and practice exams.* Rockland: Elsevier.

Suggested Readings

1. Graham, S., & Shah, S. (2003). *Linux administration: a beginner? s guide*. New York: John Wiley & Sons.
2. Desmond, B., Richards, J., Allen, R., & Lowe-Norris, A. G. (2008). *Active directory: designing, deploying, and running active directory*. Farnham: O'Reilly Media, Inc.

ITCC-6103 Web System and Technologies 3(3+0)

Today social medias are become our part of daily life and billions of common peoples using web technologies overall the world. At 1990, Tim Berners-Lee who introduced World Wide Web (www) wrote software for browser and server using HTML (Hyper Text Markup Language). Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). The three core languages that make up the World Wide Web are HTML, CSS, and JavaScript. In the IT world, the internet is an essential platform, whether it's for developing or for consumer use. When developing a website, typically three main languages come into play. This course will give an overview of Web Systems and Technologies. Students will learn the essential skills of website management; understanding of the basic Internet technology concepts, develop a prototype of interactive World Wide Web applications.

Contents

1. Overview of WWW, Web Pages, Web Sites, Web Applications,
2. TCP/IP, TCP/IP Application, Services, Web Servers.
3. WAMP Configuration.
4. Introduction to HTTP, HTML.
5. HTML5 Tags.
6. Dynamic Web Content.
7. CSS and CSS3. Client-Side Programming.
8. JavaScript: Basics, Expressions and Control Flow, Functions, Objects, and Arrays, Accessing CSS from JavaScript.
9. Form Handling. Server-Side Programming:
10. Programming in PHP, Introduction MySQL, MySQL Functions.
11. Accessing MySQL via php Mydman.
12. Cookies, Sessions, and Authentication.
13. Introduction to XML, Ajax, jQuery.
14. Browsers and the DOM.
15. Designing a Social Networking Site.

Recommended Texts

1. Nixon R, Media O'. (2012). *Learning php, mysql, javascript, and cssc, a step-by-step guide to creating dynamic websites*. Boston: Prentice Hall.

Suggested Readings

1. Jeffrey C. Jackson. (2006). *Web technologies: a computer science perspective*: New York: Prentice Hall.
2. Kumar Roy U. (2011). *Web technologies*: Oxford: Oxford University Press.
3. Shklar L, Rosen R. (2000). *Web application architecture: principles, protocols and practices*. Kansas: John Wiley and Sons.

ITCC-6203 **Virtual Systems and Services** **4(3+1)**

Virtualization technology enables a single PC or server to simultaneously run multiple operating systems or multiple sessions of a single OS. A machine with virtualization software can host numerous applications, including those that run on different operating systems, on a single platform. The host operating system can support a number of virtual machines, each of which has the characteristics of a particular OS. The solution that enables virtualization is a virtual machine monitor (VMM), or hypervisor. A hypervisor (or virtual machine monitor, VMM) is computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor runs one or more virtual machines is called a host machine, and each virtual machine is called a guest machine. This course will investigate the current state of virtualization in computing systems. Virtualization at both the hardware and software levels will be examined, with emphasis on the hypervisor configurations of systems such as VMware, Zen or Hyper-V. Processor and peripheral support for virtualization will also be examined, with a focus on emerging hardware features and the future of virtualization.

Contents

1. Overview of virtualization technology and its application, Comparison of traditional and virtual systems
2. Overview of Intel x86 platform
3. Parallel and distributed systems
4. Types of virtualization, Virtualization at Software and Hardware level
5. Virtual Machines and configuration of VMs: Managing CPU, storage, networking etc for VMs
6. Key features of VMs, Hypervisors and Configuration of VMware
7. Hyper-V and Xen hypervisors
8. Types of hypervisors: Type-1 and Type-2 hypervisor
9. Features and limitations of hypervisors
10. Para-virtualization and Para-virtualized software components
11. VMware ESXi, Xen and Microsoft virtualization implementation in the context of datacentres (lab sessions)
12. Virtualization in Cloud Computing
13. Virtualization in IoT
14. Virtualization security: security at hypervisor level, VM security
15. Future of Virtualization
16. Semester project

Pre-Requisite: Nil

Recommended Texts

1. Wolf, C., & Halter, E. M. (2006). *Virtualization: from the desktop to the enterprise*. New York: Apress.
2. Chisnall, D. (2008). *The definitive guide to the xen hypervisor*. Boston: Pearson Education.

Suggested Readings

1. Finn, A., Lownds, P., Luescher, M., & Flynn, D. (2013). *Windows server 2012 hyper-v installation and configuration guide*. Hoboken: Sybex.

ITCC-6102 **Information Technology Infrastructure** **3(3+0)**

Information Technology infrastructure: provides platform for supporting all information systems in the business Information Technology infrastructure is the combination of hardware, software, network and human resources that allow an organization to deliver information technology services to people within an organization. An organization needs desktop computers, servers, routers, switches and other equipment. Hardware is useless without software. The term Information Technology infrastructure is defined as a combined set of hardware, software, networks, facilities, etc. (including all of the information technology related equipment) used to develop, test, deliver, monitor, control, or support Information Technology services. The back-end of an Information Technology infrastructure can be split into three main elements: network, storage and computing. A traditional infrastructure has its network, storage, and computing all managed and connected within the business and consists of a lot more hardware. This course will give an overview of Information technology architectures including software systems, hardware, operating systems, databases, component technologies, networking, and architecture patterns.

Contents

1. Introduction and Definition of IT Infrastructure.
2. Non-functional Attributes; introduction, non-functional requirements, Availability Concepts, Calculating availability, Sources of Unavailability, Availability Patterns.
3. Performance concepts: introduction, Perceived performance, Performance during Infrastructure Design, Performance of a running system, performance pattern, Sources of Performance Metrics, Performance Pattern.
4. Security Concepts: Risk Management, Cryptography, Computer Crime, Security Patterns.
5. Data centres: Introduction & History, Building Blocks, Data centre Availability.
6. Networking: Building Blocks, Network virtualization
7. Network Availability, Network Performance, Network Security
8. Storage: Introduction and History, Building blocks, Availability, Performance, Security.
9. Compute: Introduction, Building Blocks, Availability, Performance, Security
10. Operating Systems: Introduction, Building Blocks, Implementing Various OSs, OS availability, OS Performance, OS Security.
11. End User Devises: Introduction & History, Building Blocks, Desktop virtualization, Device Availability, Performance, Security
12. Infrastructure life cycle
13. Infrastructure deployment options: Introduction, Hosting options, Enterprise infrastructure deployment, Software defined data-centre, (Hyper) Converged Infrastructure, Cloud computing, infrastructure as a code
14. Purchasing Infrastructure and Services
15. Deploying the Infrastructure
16. Maintaining the Infrastructure, Deploying applications

Recommended Texts

1. Laan, S. (2017). *IT infrastructure architecture-infrastructure building blocks and concepts,* (3rd ed.). London: Lulu Publisher

Suggested Readings

1. Hausman, K. K., & Cook, S. L. (2010). *IT architecture for dummies*. New York: John Wiley & Sons.
2. Britton C, Bye P (2004). *IT architectures and middleware: strategies for building large, integrated systems*. Boston: Addison-Wesley.

ITCC -6101 **Cyber Security** **3 (3+0)**

Cyber security is the protection of Internet-connected systems, including hardware, software, and data from cyber-attacks. It is made up of two words one is cyber and other is security. Cyber is related to the technology which contains systems, network and programs or data. In this course, students will be introduced to real-world cyber security challenges that organizations face, and learn to apply knowledge and skills gained through other Computer Science courses to address them. The challenges will be examined both from the attacker's perspective (how systems are exploited) and the defender's perspectives (how to secure systems or respond to threats). Common attack and defense strategies for software, web applications, networks, operating systems, cryptographic systems and humans will be explored. The course will also introduce cyber security management concepts, including security operations, risk management, security engineering and security architecture, as well as provide guidance on different career paths specializing in cyber security.

Contents

1. Threat landscape: terminology, cyber security threats, keeping up to date.
2. Authentication: access control.
3. Passwords, two-factor authentication.
4. Malware: types of malware, attack vectors, preventing infection
5. Networking and communications: fundamentals.
6. Security challenges, standards
7. Cryptography: symmetric and asymmetric cryptography, applications
8. Network security: firewalls, virtual private networks.
9. Intrusion detection / prevention.
10. When your defences fail: cyber security laws.
11. recovering from attacks
12. Managing security risks.
13. Risk analysis and management.

Recommended Texts

1. Kostopoulos, G. (2017). *Cyberspace and cyber security*. Boca Raton: Auerbach Publications.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Ciampa, M. (2012). *Security+ guide to network security fundamentals*. Boston: Cengage Learning.
2. Nguyen, N. (2018). *Essential cyber security handbook*. Nguyen, Nam: Kindle Store.

ITCC-6201 **Database Administration & Management** **4(3+1)**

Database administration refers to the whole set of activities performed by a database administrator to ensure that a database is always available as needed. Other closely related tasks and roles are database security, database monitoring and troubleshooting, and planning for future growth. Database administration is an important function in any organization that is dependent on one or more databases. Database administration is the function of managing and maintaining database management systems (DBMS) software. As such, corporations that use DBMS software often hire specialized information technology personnel called database administrators. The database administrator (DBA) is usually a dedicated role in the IT department for large organizations. However, many smaller companies that cannot afford a full-time DBA usually outsource or contract the role to a specialized vendor, or merge the role with another in the ICT department so that both are performed by one person. In Database Administration Course students gain a comprehensive overview of all the skills necessary to become database administrators. In addition to practicing existing recovery, backup and network plans, they develop their own. They gain hands-on experience practicing security procedures and using administrative tools.

Contents

1. Installation; SQL* Plus; Oracle Enterprise Manager; DBA Tools. Oracle Architectural Components: Oracle Server; Oracle Instance.
2. Physical Structure; SGA; Shared Pool; Library Cache; Data Dictionary Cache; Large Pool; User Process; Server Process; Background Processes.
3. Managing an Oracle Instance: Parameter File; Creating SPFILE; Oracle Managed Files; Startup and Shutdown Database; Alert Log File.
4. Background Trace File; User Trace File.
5. Creating Database and Data dictionary.
6. Managing Control Files and Redo Log Files.
7. Managing Tablespaces, Operations with Tablespaces.
8. Data File Management, Segments, Block.
9. Managing Undo Data, Undo Data Statistics: Managing Tables and Users:
10. Indexes Management, Maintaining Data Integrity, Constraints. Managing Privileges.
11. Basic Oracle Net Architecture: Types of Networks,
12. Oracle Net Services, Oracle Shared Server, Connection Manager, Oracle Net Connections.
13. Server-Side Configuration: The Listener Process; Configuring Listener, Sessions, Creating and Managing Listener.
14. Client-Side Configuration: Host Naming Method, Local Naming Method, Net Assistant
15. Usage and Configuration of Oracle Shared Server.

Recommended Texts

1. Bryla B, Loney K. (2007). *Oracle database 11g dba handbook*. New York: McGraw-Hill Osborne Media.
2. S. Mullins, C. (2012). *Database administration: the complete guide to dba practices and procedures*. Arlington: Addison-Wesley.

Suggested Readings

1. Connolly, T. M., & Begg, C. E. (2005). *Database systems: a practical approach to design, implementation, and management*. Harlow Essex: Pearson Education.
2. Loney, K. (2004). *Oracle database 10g: the complete reference*. London: McGraw-Hill/Osborne.

ITSC-5102 **Enterprise Systems** **3 (3+0)**

An enterprise system is a packaged application that supports and automates business processes and manages business data. They come with pre-implemented and customizable modules that reflect best practice for common business operations. The course provides insights into theoretical concepts and current practice of enterprise systems. There are four major enterprise applications: enterprise systems, supply chain management systems, customer relationship management systems, and knowledge management systems. Enterprise applications are typically designed to interface or integrate with other enterprise applications used within the organization, and to be deployed across a variety of networks (Internet, Intranet and corporate networks) while meeting strict requirements for security and administration management. The course is designed to develop students' knowledge about enterprise systems and to provide hands-on experience with enterprise systems applications. Through lectures and seminars, students learn about opportunities, challenges and approaches to enterprise systems implementation and use, and develop critical thinking skills. In addition, the students will gain practical experience in enterprise systems applications through laboratory work.

Contents

1. Introduction to Enterprise system management.
2. Business Process Management and system integration.
3. Architecture and Platform of Enterprise systems.
4. Enterprise Systems and development lifecycle technology.
5. Enterprise Systems and Business Process Reengineering, implementation and strategies.
6. Software and vendor selection.
7. Operation and post-implementation.
8. Program and project management.
9. Global, Ethics.
10. Security management.
11. Supply chain management.
12. Customer Relationship management and its factor.

Recommended Texts

1. Motiwalla, L. F, Thompson, *J. (2011). Enterprise systems for management* (2nd ed.). London: Pearson.
2. Bradford, M. (2015). *Modern ERP: select, implement, and use today's advanced business systems*. North Carolina: Lulu.

Suggested Readings

1. Weske, M. (2007). *Business process management–Concepts, languages, architectures, verlag*. Berlin: Springer-Verlag;
2. Yvonne, L. A., Martin, B., Tony, B., Bruce, D. D., Jason, F., Daniel, J. M... & Robyn, L. R. (2009). *Business process management common body of knowledge*. New Jersey: Create Space Independent Publishing Platform.

ITSC-6101 **Operations Research** **3 (3+0)**

Operations Research (OR) is a discipline that helps to make better decisions in complex scenarios by the application of a set of advanced analytical methods. It couples theories, results and theorems of mathematics, statistics and probability with its own theories and algorithms for problem solving. The field of operations research provides a more powerful approach to decision making than ordinary software and data analytics tools. Employing operations research professionals can help companies achieve more complete datasets, consider all available options, predict all possible outcomes and estimate risk. Additionally, operations research can be tailored to specific business processes or use cases to determine which techniques are most appropriate to solve the problem. The main types of scientific model are visual, mathematical, and computer models. Visual models are things like flowcharts, pictures, and diagrams that help us educate each other. The objective is to reduce decisions to a mathematical model, and identify one or more "optimal" solutions. This course aims to introduce quantities methods and techniques for effective decisions making.

Contents

1. Introduction to Model Building: An Introduction to Modeling. The Seven-Step Model-Building Process. Over view of Matrices and Vectors. Matrices and Systems of Linear Equations. The Gauss-Jordan Method for Solving Systems of Linear Equations. Linear Independence and Linear Dependence. The Inverse of a Matrix. Determinants.
2. Introduction to Linear Programming: The Graphical Solution of Two-Variable Linear Programming Problems.
3. A Work-Scheduling Problem. A Capital Budgeting Problem.
4. The Simplex Algorithm and Goal Programming: How to Convert an LP to Standard Form. The Simplex Algorithm. Using the Simplex Algorithm to Solve Minimization Problems. Solving LPs with Spreadsheets.
5. Sensitivity Analysis: An Applied Approach: A Graphical Introduction to Sensitivity Analysis. The Computer and Sensitivity Analysis. Managerial Use of Shadow Prices.
6. Sensitivity Analysis and Duality: A Graphical Introduction to Sensitivity Analysis. Some Important Formulas.
7. Sensitivity Analysis. Sensitivity Analysis When More Than One Parameter is changed.
8. Duality and Sensitivity Analysis.
9. Transportation and Transshipment Problems: Formulating Transportation Problems.
10. Finding Basic Feasible Solutions for Transportation Problems.
11. The Transportation Simplex Method. Sensitivity Analysis for Transportation Problems.
12. Network Models: Basic Definitions. Shortest Path Problems. Maximum Flow Problems.
13. CPM and PERT. Minimum Cost Network Flow Problems.
14. Minimum Spanning Tree Problems. The Network Simplex Method.

Pre-Requisite: Nil

Recommended Texts

1. Winston, W. L., & Goldberg, J. B. (2004). *Operations research: applications and algorithms* (Vol. 3). Belmont: Thomson Brooks/Cole.

Suggested Readings

1. Carter, M., Price, C. C., & Rabadi, G. (2018). *Operations research: a practical introduction*. Boca Raton: CRC Press.

ITSC-5103 Object Oriented Analysis and Design 3 (3+0)

Object-oriented analysis and design (OOAD) are a technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality. Unified Modeling Language (UML) is a general-purpose modeling language. Unified modeling language (UML) provides a standardized set of tools to document the analysis and design of a software system. Unified Modeling Language (UML) is fundamentally based on an object-oriented technique known as use case modeling. A use case model describes what a system does without describing how the system does it. In this course, the students will learn how to produce detailed object models and designs from system requirements; use the modeling concepts provided by UML; identify use cases and expand into full behavioral designs; expand the analysis into a design ready for implementation and construct designs that are reliable.

Contents

1. Principles of Object Technology: Introduction to Object Technology, Principles of modeling, and Principles of Object Orientation.
2. Introduction to UML, Unification, UML Diagrams, Unified Process & Rational Unified Process.
3. RUP Disciplines, Case Study Analysis and Basics, Case Study, About Inception, Feasibility and Risk Analysis.
4. Understanding Requirements, Requirements Types, Use Case modeling: Use Case Writing Styles, EBP Guidelines.
5. System Use Case Diagram, Use Case Table, Activity Diagram, Supplementary Specifications.
6. Vision Document, Glossary, Rational Rose Overview, Use Case & Activity Diagram Modeling in Rational Rose.
7. Elaboration Phase of RUP; System Sequence Diagram, Domain Model: Identifying Business Classes, Associations, Attributes.
8. Implementation of System Sequence & Domain Model: Use Case Operational Contracts, Business Sequence, Analysis Sequence & Collaboration Diagrams.
9. Use Case Dependencies. Analysis Use Case Diagram, Implementation of Sequence, Collaboration, Analysis Use Case Diagram.
10. State Chart Diagrams and Implementation
11. Design Patterns: GRASP: Information Expert, Creator, Cohesion & Coupling, Controller.
12. Use Case Realization Using GRASP Patterns, Design Model: Determining Visibility.
13. Modeling Generalization, Creating Design Class Diagram, Mapping Data Model to Domain Model.
14. Implementation of Design Class Diagram, Coding patterns, Mapping Design to Code.
15. More Patterns for Assigning Responsibilities, Polymorphism, Pure Fabrication, Indirection, Protected Variation. GoF Design Patterns: Adapter, Factory.
16. Gof: Singleton, Strategy, Composition, Façade and Discuss Remaining Patterns.

Pre-Requisite: Nil

Recommended Texts

1. Larman, C. (2012). *Applying UML and patterns: an introduction to object-oriented analysis and design and interactive development*. New Delhi: Pearson Education.

Suggested Readings

1. Conaway, C. F., & Page-Jones, M. (2000). *Fundamentals of object-oriented design in UML*. London: Addison-Wesley Professional.

2. Booch, G. (2005). *The unified modeling language user guide*. Arlington: Addison Wesley.

ITSC-6102 **Optimization Techniques** **3 (3+0)**

Optimization is an important tool in making decisions and in analyzing physical systems. In mathematical terms, an optimization problem is the problem of finding the best solution from among the set of all feasible solutions. An optimization algorithm is a procedure which is executed iteratively by comparing various solutions till an optimum or a satisfactory solution is found. The purpose of optimization Technique is to achieve the “best” design relative to a set of prioritized criteria or constraints. These include maximizing factors such as productivity, strength, reliability, longevity, efficiency, and utilization. This decision-making process is known as optimization. A program may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer input/output operations. The course provides students an exposure to solving non-linear optimization problems by various techniques, with due emphasis on their mathematical rigor in terms of their derivation / justification.

Contents

1. Preliminaries: Review of the theory of maxima, minima (two variables); positive definite matrices, convexity of regions and functions.
2. Quadratic function and Hessian matrix; uniqueness of minimum.
3. Classical methods for functions of one variable and n variables, Newton’s method,
4. Unconstrained Optimization.
5. Search methods for functions of one variable: Single search techniques: Bracketing method.
6. Quadratic and cubic interpolation; Fibonacci search; Golden-section.
7. Search methods for functions of n variables: method of Hooke and Jeeves, Nelder and Mead’s Method.
8. Gradient methods: Davidon-Fletcher-Powell (DFP); Fletcher- Reeves, conjugate- gradient and direct- search methods.
9. Newton’s method, method of steepest descent, Constrained Optimization.
10. Review of Lagrange multipliers technique with equality constraints.
11. Inequality constraints and slack variables; Kuhn-Tucker conditions.
12. Search methods: modified Hooke and Jeeves, the Complex method.
13. Penalty-function approach to constrained optimization.
14. Equality and inequality constraints, SUMT method of Fiacccco and McCormick.

Pre-Requisite: Nil

Recommended Texts

1. Bunday, B. (1984). *Basic optimization methods, edward arnold*. London: Bedford square.

Suggested Readings

1. Wainwright, K. (2005). *Fundamental methods of mathematical economics*. New York: McGraw-Hill.

ITSC-5104 **Digital Logic Design** **3 (3+0)**

Digital Logic is the basis of electronic systems, such as computers and cell phones. Digital Logic is rooted in binary code, a series of zeroes and ones each having an opposite value. This system facilitates the design of electronic circuits that convey information; including logic gates Digital logic design is a system in electrical and computer engineering that uses simple number values to produce input and output operations. There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR. The AND gate is so named because, if 0 is called "false" and 1 is called "true," the gate acts in the same way as the logical "and" operator. Digital logic design is used to develop hardware, such as circuit boards and microchip processors. This hardware processes user input, system protocol, and other data in navigational systems, cell phones, or other high-tech systems. As a digital design engineer, you may assist in developing cell phones, computers, and related personal electronic devices. In this course, the students will learn the basic concepts of digital circuits and their logical designs.

Contents

1. Number Systems, Operations, and Codes.
2. Logic Gates: The Inverter, The AND Gate, The OR Gate, The NOR Gate etc.
3. Boolean Algebra and logic Simplification: Boolean Operations and Expressions.
4. The Kamaugh Map, Karnaugh Map SOP Minimization, Karnaugh Map POS Minimization, Five-Variable Karnaugh Maps.
5. Combinational logic Analysis: The Universal Property of NAND and NOR Gates.
6. Combinational Logic Using NAND and NOR Gates.
7. Functions of Combinational logic: Basic Adders, Look-Ahead Carry Adders, Comparators, Decoders, Encoders.
8. Latches, Flip-Flops, and Timers: Flip-Flop Operating Characteristics, Flip-Flop Applications.
9. Counters: Asynchronous Counter Operation, Synchronous Counter Operations.
10. Shift Registers: Basic Shift Register Functions.
11. Memory and Storage: Basics of Semiconductor Memory.
12. Random-Access Memories (RAMs), Read-Only Memories (ROMs).
13. Programmable Logic: FPGA.

Pre-Requisite: Nil

Recommended Texts

1. Floyd, T. L. (2000). *Digital fundamentals*. New Delhi: Prentice Hall.

Suggested Readings

1. Floyd, T. L. (2014). *Digital fundamentals: a systems approach*. New Jersey: Pearson Education Limited.

ITSC-6103 **Design and analysis of algorithms** **3 (3+0)**

An algorithm is a set of steps of operations to solve a problem performing calculation, data processing, and automated reasoning tasks. An algorithm is an efficient method that can be expressed within finite amount of time and space. In other words, an Algorithm is a sequence of steps to solve a problem. The design of algorithms is part of many solution theories of operation research, such as dynamic programming and divide-and-conquer. Design and Analysis of Algorithm is very important for designing algorithm to solve different types of problems in the branch of computer science and information technology. The objective of the course is to teach techniques for effective problem solving in computing. The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problem. In each case emphasis of the algorithm will be used to show the efficiency of the algorithm over the native techniques.

Contents

1. Role of Algorithms in Computing.
2. Divide-and-Conquer.
3. Recurrences.
4. Sorting and Order Statistics.
5. Sorting in Linear Time.
6. Medians and Order Statistics.
7. Red-Black Trees.
8. Dynamic Programming.
9. Greedy Algorithms.
10. Elementary Graph Algorithms.
11. Single-Source Shortest Paths.
12. All-Pairs Shortest Paths.
13. Maximum Flow.
14. String Matching.

Pre-Requisite: Nil

Recommended Texts

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Boston: MIT press.
2. Levitin, A, (2007). *Introduction to the design & analysis of algorithms*. Boston: Addison Wesley.

Suggested Readings

1. Sedgewick, R. (1995). Algorithms in C++, 8th print. Boston: Addison-Wesley
2. Sedgewick, R. (2002). A*lgorithms in java.* Boston: Addison-Wesley Professional.

ITEC-6101 **Mobile Application Development** **3(3+0)**

Mobile application development is the set of processes and procedures involved in writing software for small, wireless computing devices. Like Web application development, mobile application development has its roots in more traditional software development. Android software development is the process by which new applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. For a user, the important of mobile application development is of great importance. Mobile applications send information to their customers using notification. This course is a graduate-level course of Mobile Application Development. The course aims to elaborate foundation of Mobile Application Development as a subject. It focuses on both mainstream and critical approaches to visualize and examining how these topics conceptualize Mobile Application Development as a field of study. The course explicitly relates Mobile Application Development as cognate.

Contents

1. What is Android? Obtaining the Required Tools, Installing and Configuring the Android SDK Manager.
2. Creating Your First Android Application, Anatomy of an Android Application.
3. The Big Picture, How to Get Started, Your First Android Project, A bit About Eclipse, Enhancing Your First Project.
4. Understanding Activities, Linking Activities Using Intents, Fragments, Calling Built-In Applications Using Intents, Displaying Notifications.
5. Understanding the Components of a Screen, Adapting to Display Orientation, Managing Changes to Screen Orientation.
6. Utilizing the Action Bar, Creating the User Interface Programmatically and Listening for UI Notifications.
7. Using Basic Views, Using Picker Views, Using List View to Display Long Lists, Understanding Specialized Fragments.
8. Using Image Views to Display Pictures, Using Menus with Views, Analog Clock and Digital Clock Views.
9. Saving and Loading User Preferences, Presisting Data to Files, Creating Databases.
10. Sharing Data in Android, using a Content Provider, Creating Your Own Content Providers, Using the Content Provider.
11. Sending SMS Messages Programmatically, Getting Feedback after Sending a Message, Sending SMS Messages Using Intent, Receiving SMS Messages, Sending E-mail.
12. Displaying Maps, Getting Location Data, Monitoring a Location.
13. Consuming Web Services Using HTTP, Accessing Web Services Using the Get Method, Consuming JSON Services, Sockets Programming.
14. Creating Your Own Services, Establishing Communication between a Service and an Activity, Binding Activities to Services, Understanding Threading.

Recommended Texts

1. Lee, W. M. (2012). *Beginning android 4 applications development*. Indianapolis: John Wiley & Sons.
2. Allen, G., Gopal, N., & Thomas, M. (2012). *Beginning android 4*. New York: Apress.

Suggested Readings

1. Zechner, M., & Green, R. (2011). *Beginning android 4 games development*. New York: Apress.

ITEC-6109 **E-commerce Application Development** **3 (3+0)**

Electronic commerce is the buying and selling of product, services or information via computer networks, mainly the Internet. Electronic collection of real or virtual resources, which may also be available elsewhere. For the activation of the usage of electronic commerce technologies in international trade, it is appropriate to apply the proposed model which helps to evaluate efficiency when electronic commerce technologies are used by the sellers of goods, which at the same time can be stable business partners. In the model various levels of economic evaluation are released, the costs of implementation and the usage of e-commerce technologies. The course is aimed at creating robust enterprise applications using J2EE technologies that allows for rapid change and growth. At the end of the course the students will be able to understand the concepts and standards related to the discipline of E-Commerce and analyze complex real-world problems found in electronic Commerce.

Contents

1. Object-Oriented Programming Review, Software Architectures Overview: Desktop, File/Server, 2-Tier Client/Server, Multi-Tier Client/Server.
2. Challenges of Enterprise Application Development, the Platform for Enterprise Solutions, J2EE Scenarios. J2EE Platform Technologies
3. Component Technologies, Platform Roles, Platform Services.
4. Service Technologies, Communication Technologies.
5. The Client Tier: Client Considerations, General Design Issues and Guidelines, Design Issues and Guidelines for Browser Clients, Design Issues and Guidelines for Java Clients.
6. The Web Tier: The Purpose of the Web Tier, Web-Tier Technologies.
7. Web-Tier Application Structure, Web-Tier Application Framework Design.
8. The Enterprise JavaBeans Tier: Business Logic and Business Objects, Enterprise Beans as J2EE Business Objects, Remote and Local Client Views
9. Entity Beans, Session Beans, Message-Driven Beans, Design Guidelines, Portability Guidelines.
10. Integrating with the Enterprise Information System Tier
11. Integration Scenarios, J2EE Integration Technologies, Application Integration Design Approaches
12. Developing an Integration Layer. Packaging and Deployment

Pre-Requisite: Nil

Recommended Texts

1. Singh, I., Johnson, M., & Stearns, B. (2002). *Designing enterprise applications with the j2ee platform*. Boston: Addison-Wesley Professional.

Suggested Readings

1. Sriganesh, R. P., Brose, G., & Silverman, M. (2006). *Mastering enterprise javabeans 3.0*. Indianapolis: John Wiley & Sons.
2. Mark, C. (2010). *Sun certified enterprise architect for java ee study guide.* Delhi: Pearson Education.

ITEC-6102 **Mobile and Wireless Network** **3(3+0)**

The definition of mobile and wireless varies from person to person and organization to organization. In many cases, the terms mobile and wireless are used interchangeably, even though they are two different things. Let's start with the term mobile. Mobile is the ability to be on the move. A mobile device is anything that can be used on the move, ranging from laptops to mobile phones. As long as location is not fixed, it is considered mobile. Areas that are not included in our definition of mobile include remote offices, home offices, or home appliances. While these are definitely remote, they are not considered mobile. A wireless network is a computer network that uses wireless data connections between network nodes. In this course student should be able to perform a basic design of a wireless system for both coverage and capacity for given performance targets. They should also be able to evaluate the coverage, capacity and/or performance of a wireless system. Practical engineering constraints and requirements including radio and telecom traffic performance targets are included in this approach, providing the network performance objective for the design.

Contents

1. Basics of Wireless Local Area Networks
2. Radio Transmitters and Receivers, Multiple Access Methods.
3. Radio Propagation.
4. Antennas and Transmission Lines.
5. Communication Protocols and Modulation.
6. High-Speed Wireless Data.
7. GSM/Cellular Networks.
8. Security in Wireless Local Area Networks.
9. Voice Over Wi-Fi and Other Wireless Technologies
10. Mobile Ad Hoc Networks.
11. Wireless Sensor Networks.
12. Reliable Wireless Networks for Industrial Applications.
13. Industrial Applications of Wireless Mesh Networks.
14. Applications and Technologies.

Pre-Requisite: Nil

Recommended Texts

1. Chandra, P., Dobkin, D. M., Bensky, D., Olexa, R., Lide, D., & Dowla, F. (2007). *Wireless networking: know it all*. New York: Newnes.

Suggested Readings

1. Stallings, W. (2009). *Wireless communications & networks*. Delhi: Pearson Education.
2. Smith, C. (2014). *Wireless networks: design and integration for lte, evdo, hspa and wimax*. New York.: McGraw-Hill Education.

ITEC-6103 **Cloud Computing** **3(3+0)**

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud is just a metaphor for the Internet. The cloud is also not about having dedicated network attached storage (NAS) hardware or server in residence. An example of a Cloud Computing provider is Google's Gmail. Gmail users can access files and applications hosted by Google via the internet from any device. Mainstream cloud infrastructure services and related vendor solutions are also covered in detail. The course also covers the cloud security model and associated challenges and delves into the implementation and support of high-performance computing and Big Data support capabilities on the cloud.

Contents

1. Distributed systems, Characteristics, Design goals.
2. Types of distributed systems.
3. What is Cloud Computing? Different perspectives, Properties and characteristics, Benefits
4. Service and deployment models of Cloud computing.
5. Service models: IaaS, PaaS, SaaS
6. From IaaS to PaaS, PaaS and SaaS properties, Issues, characteristics and Implementation
7. Modern On-Demand Computing, Amazon’s Elastic Cloud
8. Amazon EC2 Service, Characteristics, Amazon, Simple DB
9. Amazon Simple Queue Service (Amazon SQS).
10. Amazon, Cloud Front
11. Amazon Elastic Block Store (EBS) Virtualization
12. From emulation to virtualization.
13. Goals of virtualization.

Recommended Texts

1. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. New York: CRC press.
2. Kshemkalyani, A. D., & Singhal, M. (2011). *Distributed computing: principles, algorithms, and systems*. New York: Cambridge University Press.

Suggested Readings

1. Rothon, J. (2009). *Cloud computing explained: implementation handbook for enterprises.* London: Recursive Press.
2. Linthicum, D. S. (2009). *Cloud computing and SOA convergence in your enterprise: a step-by-step guide*. Boston: Addison-Wesley.
3. Reese, G. (2009). *Cloud application architectures: building applications and infrastructure in the cloud*. California: O'Reilly Media, Inc.

ITEC-6104 **Internet of Everything** **3 (3+0)**

The aim of the Internet of Things Value Creation Network is to coordinate and help to increase and optimize the utilization of results and value creation in the area of IOT. The Internet of Things (IoT) involves the internet-connected devices we use to perform the processes and services that support our way of life. Another component set to help IoT to succeed is cloud computing, which acts as a sort of front end.Create the liaison to the European Technology Platforms addressing IoT activities primary Epos’s, e Mobility and ENIAC and ARTEMIS to cover innovation strategies for the development of enabling technologies (Nano-electronics, embedded systems, communication technologies, software and cloud computing, etc.) required for IoT applications. The goal of this course is to introduce you to fundamental concepts and technologies that enable the IoE and help you understand its benefits as well as potential risks. The course presents introductory material and is intended to be easily completed by anyone with a basic appreciation of computer technologies. By completing this course, you will not become an IoE expert, but you will become an informed individual.

Contents

1. Internet of Things.
2. IoT infrastructures.
3. Device/Cloud Collaboration Framework for Intelligence Applications IoT
4. Communication Protocols for IoT, Network Layers, Transport and Application layer
5. Fog Computing.
6. Programming Frameworks for Internet of Things.
7. Virtualization on Embedded Boards as Enabling Technology for the Cloud of Things.
8. Micro Virtual Machines (Micro VMs) for Cloud-Assisted Cyber-Physical Systems.
9. Design and Implement Scalable, Flexible, and open IoT solutions using Web technologies
10. IoT data management and Analytics.
11. A Framework for Distributed Data Analysis for IoT
12. Security and Privacy in the Internet of Things.
13. Devices in the Internet of Things Internet of Things Applications.

Cloud-Based Smart-Facilities Management.

Recommended Texts

1. Buyya, R., & Dastjerdi, A. V. (Eds.). (2016). *Internet of things: principles and paradigms*. San Francisco: Elsevier.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Somani, A. K., & Deka, G. C. (Eds.). (2017). *Big data analytics: tools and technology for effective planning*. Florida: CRC Press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

ITEC-6105 **Data Warehousing** **3(3+0)**

This course gives an introduction to methods and theory for development of data warehouses and data analysis using data mining. Data quality and methods and techniques for preprocessing of data. Modeling and design of data warehouses. Algorithms for classification, clustering and association rule analysis. Practical use of software for data analysis. To learn concepts and skills for designing data warehouses and creating data integration workflows. Introduction of tools for data warehousing. It will address the opportunities and challenges of applying data mining techniques in academics, industry, businesses, sciences and the Web. Several aspects of the data mining process are covered in this course such as: data gathering and storage, data selection and preparation, model building and testing, results interpretation and validation and models application. Finance professionals wanting to learn about data warehouses for reporting and analysis purposes, businesses looking to update and improve on data storage, analysis, and reporting processes.

Contents

1. Introduction to Data Warehousing.
2. Data Warehouse System Lifecycle.
3. Analysis and Reconciliation of Data Sources: Inspecting and Normalization Schemata, Integration Problems, Integration Phases, Defining Mapping.
4. User Requirement Analysis: Interviews, Glossary-based Requirement Analysis, Additional Requirements.
5. Dimensional Fact Model, Events and Aggregation, Temporal Aspects, Overlapping Fact Schemata, Formalizing the Dimensional Fact Model.
6. Conceptual Design: ER Schema-based Design, Relational Schema-based Design.
7. Workload and Data Volume
8. Logical Modeling.
9. Logical Design.
10. Data-staging Design: Population Reconciled Databases, Cleansing Data, Populating Dimensional Tables, Populating Fact Tables, Populating Materialized View
11. Indexes for the Data Warehouse.
12. Physical Design: Optimizers, Index Selection, splitting a Database into Tablespaces, Allocating Data Files, Disk Block Size.
13. Data Warehouse Project Documentation
14. Case Studies, Tools for Data Warehousing: MS SQL and Teradata

Recommended Texts

1. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., & Becker, B. (2008). *The data warehouse lifecycle toolkit*. New York: John Wiley & Sons.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Ponniah, P. (2011). *Data warehousing fundamentals for its professionals*. New York: John Wiley & Sons.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

ITEC-6110 **Semantic Web Techniques** **3(3+0)**

The Semantic Web is a W3C Activity for representing information in the World Wide Web in a machine-readable fashion: such that it can be used by machines not just for display purposes, but for automation, integration, and reuse across applications. This course introduces techniques that are useful stand-alone and can be integrated for building a semantic web. In this course students will be introduced to the Semantic Web vision, as well as, the languages and tools useful in Semantic Web programming. They will understand how this technology revolutionizes the World Wide Web and its uses. Ontology languages (RDF, RDF-S and OWL) and technologies (explicit metadata, ontologies, logic, and inference) will be covered. In addition, students will be exposed to; ontology engineering, application scenarios, Semantic Web Query Languages, Description Logic and state of the art Semantic Web applications, such as linked data development. Student will also learn how to develop semantic applications with Java and Jena APIs.

Contents

1. Introduction to the semantic web.
2. Structured Web Documents – XML, RDF
3. Introduction to ontologies.
4. Ontology Engineering
5. Ontology Engineering
6. Ontology languages for the semantic web.
7. Resource Description Framework (RDF).
8. Lightweight ontologies: RDF Schema.
9. Web Ontology Language (OWL).
10. Query language for RDF: SPARQL.
11. Description Logic
12. Building Semantic Web Applications (Apache Jena Framework)
13. Building Semantic Web Applications
14. Semantic Web Applications (E-learning, Web services)
15. Ontology Engineering (Protégé OWL API)
16. Semantic web and Web 2.0
17. Applications of Semantic Web.

Recommended Texts

1. Cardoso, J., Sheth, A., & Yu, L. (2009). *Semantic web services, processes and applications.* New York: Springer

Suggested Readings

1. Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). *Foundations of semantic web technologies*. Florida: CRC press.

ITEC-6106 **Knowledge Management** **3 (3+0)**

Knowledge, knowledge products, and knowledge processes are the key ingredients of productivity and profitably in the business world. Knowledge revolution has given birth to knowledge economies. Knowledge management (KM) is the field of managing human knowledge. Knowledge Management is commonly associated with processes like knowledge creation, knowledge sharing, knowledge storage, knowledge refinement, etc. Knowledge management (KM) is an area that has captured the attention of many organizations that are concerned with the way's knowledge is managed more effectively. KM offers systematic methods in leveraging and managing organizational knowledge through KM processes of creation, storing, sharing, and application of knowledge. The need is to gain a sustainable competitive edge among partners as well as competitors by learning how to leverage intangible assets in new and creative ways. This course introduces students to appraise current thoughts on knowledge management in the light of contemporary debates on knowledge productivity, strategic capability, and organizational learning. Further, it enables them to learn how to develop, manage, and evaluate knowledge management theories, models, frameworks, systems, initiatives, and best practices.

Contents

1. History and paradigms of knowledge management
2. Types of knowledge: explicit knowledge, tacit knowledge, embedded knowledge
3. KM processes: knowledge discovery/ detection
4. KM frameworks and models: SECI, Alen Frost's, and Van Buren's models
5. Knowledge capture and codification: group, knowledge codification
6. Knowledge sharing and communities of practice: types of communities
7. Knowledge application: task analysis and modeling, knowledge reuse
8. The role of organizational culture: different types of cultures
9. Knowledge management tools: knowledge blogs, mashups, PKM
10. Knowledge management strategy: knowledge audit, gap analysis
11. The value of knowledge management: ROI and metrics
12. Organizational learning and organizational memory
13. Major categories of knowledge management roles
14. The profession and ethics of knowledge management

Recommended Texts

1. Dalkir, K. (2017). *Knowledge management in theory and practice*. Cambridge: Massachusetts Institute of Technology.

Suggested Readings

1. Tiwana, A. (2007). *The knowledge management toolkit: orchestrating It, strategy, and knowledge platforms*. New Jersey: Prentice Hall PTR.
2. Geisler, E. & Wickramasinghe, N. (2009). *Principles of knowledge management: theory, practices, and cases*. Armonk: M.E. Sharpe.
3. Jennex, M. (2008). *Knowledge management: concepts, methodologies, tools and applications.* Hershey PA: Information Science Reference.

ITEC-6107 **Network Design and Management** **3 (3+0)**

The course introduces students to the basic understandings of Computer networks, different networking devices, interconnecting devices, protocols and their importance in networks. Working and performance of key technologies in network designs. Networks on the basis of network design best practices. Matrices necessary to design best network designs. The primary objective of this course is to give students an understanding of how to design, manage and secure computer networks and systems. It further examines how network managers and security engineers can strategically use different techniques to capture, collect and analyze network and system data to create a competitive advantage. It is not uncommon to observe lack of practical knowledge in bright young graduates or even the more experienced ones, who have specialized in computer sciences and telecommunications, when they enter the work environment, Making NMD a part of academic curriculum is necessity to remedy this deficiency. The aim of the course is to give students the background they will need to attack the problems in practical environment.

Contents

1. Analyzing Business Goals and Constraints: Network Design Methodology, Analyzing Business Goals.
2. Analyzing Technical Goals and Tradeoffs: Scalability, Availability, Manageability etc.
3. Characterizing the Existing Internetwork and Infrastructure, Checking the Health of the Existing Internetwork.
4. Designing a Network Topology: Modular Network, Designing a Campus Network Design.
5. Virtual LANs, Wireless LANs, Redundancy and Load Sharing in Wired LANs.
6. Designing Models for Addressing and Numbering.
7. Selecting Switching and Routing Protocols.
8. Developing Network Security Strategies.
9. Physical Network Design: Selecting Technologies and Devices for Campus Networks.
10. Selecting Technologies and Devices for Enterprise Networks: Remote-Access etc.
11. Testing Network Design: Using Industry Tests, Building and Testing a Prototype Network.
12. Optimizing Network Design: Optimizing Bandwidth Usage with IP Multicast Technologies
13. Documenting Network Design: Responding to a Customer.

Pre-Requisite: Nil

Recommended Texts

1. Oppenheimer, P., Nabozny, K., & Wilson, J. B. (2011). *Top-down network design*. Indianapolis: Cisco Press.

Suggested Readings

1. Chao, L. (2009). *Networking systems design and development*. Indianapolis: CRC Press.
2. Karris, S. T. (2009). *Networks: design and management*. Texas: Orchard Publications.

ITEC-6108 **Business Intelligence and Analytics** **3 (3+0)**

Business Intelligence (BI) is the field of combining data, technology, business processes, and analytics to optimize business decisions and drive success. The ability to generate insights using data in today's world is crucial for any organization's success into the future. Turning data into valuable information is a necessity for ever-changing markets. This course will give an introduction to analytical tools and skills that can be used to understand, analyses, and evaluate the challenges and opportunities for an organization. Students will learn about various technical aspects of BI and understand the processes involving in planning, designing, building, and maintaining the BI environment. Further, this course will develop skills in the use and application of various techniques and tools for driving insights from data for effective business decision making. This course introduced to key data analytics concepts such as systems thinking, multi-level perspectives and multidisciplinary methods for envisioning futures, and apply them to specific real-world challenges you and your organisation may face.

Contents

1. Value drivers, performance metrics and key performance indicators
2. Use cases for BI
3. BI success factors, strategic versus tactical planning, BI strategy and plan
4. BI environment: analytics platform, frameworks, services, and systems evaluation
5. Business process and information flow
6. Data requirements analysis
7. Data warehouses and the technical BI architecture
8. Data profiling
9. Business rules
10. Data quality
11. Data integration
12. Deriving insight from data
13. Knowledge discovery & delivery
14. Installations, configuring and maintaining the BI server
15. Creating reports using answers and dashboards.

Recommended Texts

1. Rittman, M. (2013). *Oracle business intelligence 11g developer's guide*. New York: McGraw-Hill.

Suggested Readings

1. Larson, B. (2012). *Delivering business intelligence with microsoft sql server 2012*. New York: McGraw-Hill.
2. Vitt, E., Luckevich, M. & Misner, S. (2002). *Business intelligence: making better decisions faster*. Redmond, Wash: Microsoft Press.
3. Laberge, R. (2011). *The data warehouse mentor: Practical data warehouse and business intelligence insights*. New York: McGraw-Hill.

ITEC-6111 **Data Mining** **3(3+0)**

The course introduces students with basic applications, concepts, and techniques of data mining and to develop their skills for using recent data mining software to solve practical problems in a variety of disciplines. At the end of the course the students will be able to understand the applications, concepts, and techniques of data mining. Learn different data mining tools and apply basic data mining techniques to actual problems. Moreover, student will be able to identify appropriate data mining algorithms to solve real world problems and Compare and evaluate different data mining techniques like classification, prediction, clustering and association rule mining. Data mining refers to a set of techniques that have been designed to efficiently find interesting pieces of information or knowledge in large amounts of data. Association rules, for instance, are a class of patterns that tell which products tend to be purchased together. The knowledge discovery process includes data selection, cleaning, coding, using different statistical and machine learning techniques, and visualization of the generated structures.

Contents

1. Data-Mining Concepts, Data-Mining Process, Large Data Sets, Data Warehouses.
2. Preparing the Data: Raw Data- Representation, Characteristics, Transformation; Missing Data, Time-Dependent Data, Outlier Analysis.
3. Data Reduction: Dimensions of Large Data Sets, Feature Reduction, Relief Algorithm.
4. Learning Machine, SLT, Types of Learning Methods, SVMs, kNN: Nearest Neighbor Classifier, Model Selection versus Generalization, Model Estimation.
5. Statistical Inference, Assessing Differences in Data Sets, Bayesian Inference, Predictive Regression, ANOVA, Logistic Regression, Log-Linear Models, LDA.
6. Decision Trees and Decision Rules.
7. Artificial Neural Networks.
8. Ensemble Learning: Ensemble-Learning Methodologies, Combination Schemes for Multiple Learners, Bagging and Boosting, Ada Boost.
9. Clustering, Similarity Measures, Agglomerative Hierarchical Clustering, Partitional Clustering, Incremental Clustering, DBSCAN Algorithm. BIRCH Algorithm, Clustering Validation.
10. Association Rules, Web Mining and Text Mining.
11. Genetic Algorithms, Fuzzy Sets and Fuzzy Logic.
12. Visualization Methods.
13. Data Mining Tools: Weka, CBA and Yale, etc.

Pre-Requisite: Database Systems

Recommended Texts

1. Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms. Piscataway,* New Jersey Hoboken, NJ: IEEE Press Wiley.
2. Han, J., Kamber, M. & Pei, J. (2012). *Data mining: concepts and techniques.* Amsterdam Boston: Elsevier/Morgan Kaufmann.

Suggested Readings

1. Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.
2. Singh, R. & Asthana, A. (2012). *Data mining and data warehousing practical machine learning*

tools techniques. Saarbrucken: Lap Lambert Academic Publishing.

ITEC-6112 Enterprise Resource Planning Systems 3(3+0)

This course provides students with an understanding of what Enterprise Systems (also commonly-termed as Enterprise Resource Planning Systems, ERPs) are. The course is basically meant for introducing ERP Systems, which provides for integrating planning throughout the business cycle from raw-materials, shop floor control to the related human-resource and finance processes. The course discusses the role of ERP Systems and Software to reduce inventories, waste, scrap, and rework and how to utilize resources efficiently. Functional modules will be given special emphasis to make the users feel the practical world of enterprise planning and its impact on the bottom line of financial statements. This course systematically presents several conceptual and pragmatic methodologies, tools and techniques for various phases of implementation in an enterprise. After learning about what these systems are, we would touch upon why these systems are useful to companies, through which students would get to see the various jobs and positions that are associated with the use and deployment of ERPs.

Contents

1. Introduction to Enterprise Resource Planning Systems.
2. ERP Technology.
3. ERP and Business Process Reengineering.
4. Systems Diagramming and the Process Map.
5. ERP Life Cycle: Planning and Package Selection.
6. ERP Life Cycle: Implementation and Operation and Maintenance.
7. ERP Sales, CRM and Knowledge Management.
8. ERP Financials.
9. Human Capital Management
10. Self-Service and Outsourcing.
11. Manufacturing Systems and Supply Chain.
12. Auditing ERP
13. Business Intelligence
14. Performance Management.

Recommended Texts

1. Bradford, M. (2015). *Modern ERP: select, implement, and use today's advanced business systems*. Morrisville, NC: Lulu.com.
2. Olson, D. L. (2003). *Managerial issues of enterprise resource planning systems*. New York: McGraw-Hill, Inc.

Suggested Readings

1. Wagner, B., & Monk, E. (2008). *Enterprise resource planning*. Boston: Cengage Learning.
2. Srivastava, D., & Batra, A. (2010). *Erp systems*. New Delhi: IK International Publishing House.

ITEC-6113 **Network Programing** **3(3+0)**

Computer network protocols implementation using socket programming. The course covers the Intern et protocol suite (e.g. IP, TCP, UDP, ICMP, and FTP) and clientserver design (e.g. Connectionless, co nnection oriented, multiprotocol). Programming projects represent a significant component of the cou rse. The aim is to teach the students to identify and describe the purpose of each component of the TCP/IP protocol suite and learn to develop large and complex client-server application using TCP/IP. Analyze, design, install, configure, document, and troubleshoot network & system hardware and operating systems. Implement LANs using both static and dynamic addressing techniques, including subnetting. Identify risks, assess threats, and develop effective countermeasures aimed at protecting computer assets and data. Communicate effectively both orally and in writing, using proper computer system & networking terminology. Having successfully completed this course, the student will be able to demonstrate mastery of main protocols comprising the Internet. Develop skills in network programming techniques. Implement network services that communicate through the Internet.

Contents

1. Introduction and TCP/IP
2. BSD Networking, Unix standard, 64-bit Architectures
3. Transport Layer details
4. Sockets Introduction and Elementary TCP Sockets
5. TCP/IP client server Application
6. I/O Multiplexing
7. Socket Options
8. Socket states, generic, IPv4, IPv6, ICMPv6 socket options
9. Elementary UDP Sockets:
10. Elementary Name and Address Conversions.
11. IPv4 and IPv6 Interoperability.
12. Daemon Processes and advanced I/O functions.
13. Non-blocking, I/O and ioctl operations.
14. Routing Sockets, Broadcasting, Multicasting

Pre-Requisite: Nil

Recommended Texts

1. Stevens, W. R., Fenner, B., & Rudoff, A. M. (2018). *Unix network programming.* Majitar: SMIT-SMU.

Suggested Readings

1. Hart, J. M. (2010). *Windows system programming*. Boston: Pearson Education.
2. Kerrisk, M. (2010). *The Linux programming interface: a linux and unix system programming handbook*. San Francisco: No Starch Press.
3. Love, R. (2010). *Linux kernel development*. Boston: Pearson Education.

ITEC-6114 **Information Systems and Audit** **3(3+0)**

The course introduces students to understand the concepts and standards related to the discipline of Information System Audit and to analyze and Audit Information Systems. This course addresses the specific issues of how we can attest to the efficacy of our information resources and provide assurance that the objectives and performance of these systems are being met. Recent changes to legislation, the greater reliance on information resources by organizations and the increased access to technology has made auditing of this resource an imperative for all organizations. As a result of taking this course the student, should be able to examine the multiple layers of IS security in organizations. Analyze the risk management approach to information assets' security with respect to operational and organizational goals. Other advanced topics on information security such as mobile computing security, security and privacy of cloud computing, as well as secure information system development will also be discussed.

Contents

1. Introduction to Auditing, Assurance and Internal Control.
2. IT Governance and Management: Risk Management.
3. Organization Structure and Responsibilities, Auditing IT Governance.
4. The Audit Process: Audit Management, ISACA Auditing Standards.
5. Internal Controls, Performing an Audit, Control Self-Assessment.
6. IT Life Cycle Management: Business Realization, Project Management.
7. Infrastructure Development and Implementation.
8. Auditing the Software Development Life Cycle.
9. IT Service Delivery and Infrastructure, Information Systems Operations.
10. Disaster Recovery Planning, Auditing IS Infrastructure and Operations.
11. Information Asset Protection: Information Security Management.
12. Network Security Controls, Environmental Controls.
13. Over view of Popular Methodologies, Framework and Guidelines.
14. Overview of Computer-Assisted Audit Tools and Techniques.

Pre-Requisite: Nil

Recommended Texts

1. Nyirongo, A. (2015). *Auditing information systems: enhancing performance of the enterprise*. Bloomington: Trafford Publishing.
2. Weber, R. A. (1998). *Information systems control and audit*. Boston: Pearson Education.

Suggested Readings

1. Harris, S. (2009). *Cisa certified information systems auditor all-in-one exam guide*. New York: McGraw-Hill, Inc.
2. Hall, J. A. (2015). *Information technology auditing*. Boston: Cengage Learning.
3. Senft, S., & Gallegos, F. (2008). *Information technology control and audit*. Florida: CRC Press.

ITEC-6115 **Routing and Switching** **3(3+0)**

The aim is to teach the students is to understand the switching details and mechanism architecture and operating principle of router. Operations of wireless LANs, WANs Solutions and effectiveness of routing protocols and configurations. This course introduces the architecture, structure, functions, components, and models of the Internet and other computer networks. The principles and structure of IP addressing and the fundamentals of Ethernet concepts, media, and operations are introduced to provide a foundation for the curriculum. By the end of the course, participants will be able to build simple LANs, perform basic configurations for routers and switches, and implement IP addressing schemes. At the end of the course student will be able to understand and describe the devices and services used to support communications in data networks and the Internet. Understand and describe the role of protocol layers in data networks. Utilize common network utilities to verify small network operations and analyze data traffic.

Contents

1. Ethernet Basic
2. Virtual LANs and VLAN Trunking
3. IP Addressing
4. IP Services
5. IP Forwarding (Routing)
6. EIGRP: Basic States, EIGRP Convergence.
7. OSPF: OSPF Database Exchange.
8. IGP Routing
9. Fundamentals of BGP Operations
10. BGP Routing Policies
11. Wide Area Network
12. IP Multicasting
13. IP Multicast Routing

Pre-Requisite: Nil

Recommended Texts

1. Odom, W., Healy, R., & Donohue, D. (2010). *Ccie routing and switching certification guide*. Boston: Pearson Education.

Suggested Readings

1. Hartpence, B. (2011). *Packet guide to routing and switching: exploring the network layer*. New York: O'reilly media, inc.
2. Ellis, B., Uecker, J., & Means, S. (2010). *Ccie routing and switching v4. 0 quick reference*. Florida: Cisco Press.

ITEC-6116 **Business Process Management** **3 (3+0)**

This course looks at ways in which business processes can be analyzed, redesigned, and improved thus ensuring that they are meeting the needs of customers and the enterprise. A business process is a set of related activities that together realize a business goal in an organizational and technical context. These processes take place in a single organization but may need to interact with processes in other organizations. This course will introduce you to business process management. You'll learn how business processes can help you improve your company's bottom line by providing a higher level of quality and consistency for your customers. Business Process Management (BPM) is concerned with the concepts, methods, and techniques that support the design, improvement, management, configuration, enactment, and analysis of business processes that deliver lean and customer focused business processes. BPM includes process modelling that includes defining, analyzing, and improving processes. Students will be able to understand business process from a management and process analyst perspective, learn skills, analytical frameworks and general principles for managing business processes.

Contents

1. Business introduction
2. Evolution of enterprise systems architectures
3. Business process modeling
4. Process orchestrations
5. Process choreographies
6. Properties of business processes
7. Business process management architectures
8. Business process management methodology

Recommended Texts

1. Weske, M. (2012). *Business process management: concepts, languages, architectures*. Berlin New York: Springer.

Suggested Readings

1. Yvonne, L. A., Martin, B., Tony, B., Bruce, D. D., Jason, F., Daniel, J. M. & Robyn, L. R. (2009). *Business process management common body of knowledge*. Chicago: Association of Business Process Management Professionals.
2. Becker, J., Kugeler, M. & Rosemann, M. (2011). *Process management: a guide for the design of business processes*. Berlin London: Springer.
3. Jeston, J. & Nelis, J. (2008). *Business process management: practical guidelines to successful implementations*. Amsterdam Boston London: Elsevier/Butterworth-Heinemann.
4. Malik, T. (2009). *Process management: practical guidelines to successful implementations*. New Delhi: Global India Publications.

ITEC-6117 **Artificial Intelligence** **3 (3+0)**

This course will introduce the basic principles in artificial intelligence. Students will learn representation schemes, problem solving paradigms, constraint propagation, and search strategies. Explore different areas of AI application such as knowledge representation, natural language processing, expert systems, vision and robotics. At the end of the course the students will be able to: Knowledge of current progresses related to AI. Introduction of many advances' subjects taught at MS and PhD level and applied research. Artificial intelligence (AI) is a research field that studies how to realize the intelligent human behaviors on a computer. The main research topics in AI include: problem solving, reasoning, planning, natural language understanding, computer vision, automatic programming, and machine learning, and so on.

Contents

1. Introduction: What is AI, Foundations of AI, History of AI. Intelligent Agents: Agents and Environments, The Nature of Environments, The Structure of Agents
2. Problem Solving by Searching: Problem Solving Agents, Searching for Solutions
3. Breadth-First Search, Depth-First Search, Depth-limited Search, Iterative Deepening, Depth-first Search, Comparison of Uninformed Search Strategies.
4. Informed Search and Exploration: Informed (Heuristic) Search Strategies: Greedy Best-first Search, A* Search, Heuristic Functions, Local Search Algorithms and Optimization Problems.
5. Constraint Satisfaction Problems: Backtracking Search for CSPs, Local Search for CSPs. Adversarial Search: Games, Minimax Algorithm, Alpha-Beta Pruning.
6. Reasoning and Knowledge Representation: Introductions to Reasoning and Knowledge Representation, Propositional Logic, First Order Logic: Syntax and Semantics of First-Order Logic, Knowledge Engineering in First-Order Logic,
7. Inference in First-Order Logic: Inference rules for quantifiers, A first-order inference rule, Unification, Forward Chaining, Backward Chaining, A backward chaining algorithm, Logic programming, The resolution inference rule
8. Introduction to Prolog Programming
9. Reasoning Systems for Categories, Semantic Nets and Description logics, reasoning with Default Information: Open and closed worlds, Negation as failure and stable model semantic.
10. Reasoning with Uncertainty & Probabilistic Reasoning: Acting Under Uncertainty, Bayes' Rule.
11. Representing Knowledge in an Uncertain Domain, the Semantics of Bayesian Networks.
12. Learning from Observations: Forms of Learning, Inductive Learning, Learning Decision Trees
13. Knowledge in Learning, Explanation-Based Learning, Inductive Logic Programming.
14. Statistical Learning, Neural Networks

Pre-Requisite: Nil

Recommended Texts

1. Russell, S., Norvig, P. & Canny, J. (2003). *Artificial intelligence: a modern approach.* New York: Prentice Hall/Pearson Education.

Suggested Readings

1. Jones, M. (2008). *Artificial intelligence: a systems approach*. Hingham, MA: Infinity Science Press.

2. Lucci, S. & Kopec, D. (2013). *Artificial intelligence in the 21st century: a living introduction*. Dulles, VA: Mercury Learning and Information.

ITEC-6118 **Natural Language Processing** **3 (3+0)**

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes, Context-free parsing, Earley algorithm, Extending CFG.
4. Probabilistic parsing, Parsing tricks, Human sentence processing.
5. Semantics, Forward-backward algorithm, Expectation Maximization.
6. Finite-state algebra, Finite-state implementation, Finite-state tagging, Noisy channels and FSTs, More FST examples.
7. Programming with regexps, Morphology and phonology.
8. Optimal paths in graphs, Structured prediction.
9. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Pre-Requisite: Nil

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing* (3rd ed.). New York: Pearson
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python.* Massachusetts: OReilly Media Reading.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge: MIT press.

ITEC-6119 **Grammar Engineering** **3 (3+0)**

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview, LFG Basics.
3. LFG Basics II, Templates I, MacOSX, Unix.
4. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file
5. Lexical Rules, Passive and Argument alternations.
6. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.
11. Finite-State Morphology (FSM) I.
12. FSM II (-unknown), Free Word Order and the Shuffle Operator.
13. Meta-categories, Meta-rule-macros and Coordination.
14. Project

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. California: CSLI. Publications Stanford.
2. Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., & Newman, P. (2008). *Xle documentation.* California: Palo Alto Research Center.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar*. Leiden: Brill.
2. Dalrymple, M., Kaplan, R. M., Maxwell III, J. T., Maxwell, J. C., & Zaenen, A. E. (1995). *Formal issues in lexical-functional grammar*. Stanford: Center for the Study of Language (CSLI).

ITEC-6120 **Text Processing & Data Analysis** **3(3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing, Text Processing Functions.
6. Loop Control, Hashes, DBM Databases, Advanced Sorting.
7. Regular Expressions, Environment Variables, CGI-Programming.
8. Process Management, References and Data Structures.
9. Graphics, JavaScript.

Pre-Requisite: Nil

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. Sebastopol: O'Reilly & Associates, Inc.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming perl: unmatched power for text processing and scripting*. Sebastopol: O'Reilly Media, Inc.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl cookbook: solutions & examples for perl programmers*. Sebastopol: O'Reilly Media, Inc.
2. Lidie, S., & Walsh, N. (2002). *Mastering perl/tk: graphical user interfaces in perl*. Sebastopol: O'Reilly Media, Inc.

ITEC-6121 **Speech Recognition & Synthesis** **3 (3+0)**

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background: N-gram Language Modeling
2. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization, Phonetics Speech Synthesis, pages 1-10 , Optional Advanced Reading, Text Segmentation and Organisation, Text Decoding.
3. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software, Prosody Prediction from Text.
4. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis), Unit Selection Synthesis, Optional Advanced Reading.
5. ASR: Noisy Channel Model, Bayes, HMMs, Forward, Viterbi, Hidden Markov Models, Automatic Speech Recognition.
6. ASR: Feature Extraction and Acoustic Modeling, Evaluation, Speech Recognition: Advanced Topics.
7. ASR: Learning (Baum-Welch) and Disfluencies, Automatic Speech Recognition, Speech Recognition: Advanced Topics.

Pre-Requisite: Nil

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing*. New York: prentice Hal
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press.

ITEC-6122 **Big Data Analytics** **3 (3+0)**

The main aim of the course is to frame big data jobs as Apache Spark Scripts. Students should be capable enough to optimize an algorithm by partitioning it across worker nodes. At the end of this course, student will become familiar with the fundamental concepts of Big Data management and analytics. Also, they will become competent in developing distributed code using Scala language and Apache Spark API. The course will provide enough concepts to deploy a spark cluster and run jobs on the cluster. Course will also cover GraphX, MLib, SparkSQL libraries to study their application in social networks and link prediction. This course is for those new to data science. No prior programming experience is needed, although the ability to install applications and utilize a virtual machine is necessary to complete the hands-on assignments. This course focuses on RDD based algorithm design, with an emphasis on text processing algorithms common in natural language processing, information retrieval, and machine learning.

Contents

1. Big Data: Issues and challenges
2. Big Data Tools: An overview
3. Functional Programming Paradigm
4. Scala: Basics, Functions, and Data Structures
5. Scala Practical Exercise
6. Apache Spark & RDD
7. Deploy Apache Spark Cluster
8. Implement and Run Apache Spark Job on Cluster
9. Broadcast Variables and Accumulative Variables in Apache Spark
10. Island, Global, and Grid Models
11. Translating Embeddings for Modeling Multi-relational Data
12. Scalable Genetic Algorithms using Apache Spark
13. ACO for Link Prediction
14. Using Neighborhood for Link Prediction Problems
15. GraphX, MLib, SparkSQL

Pre-Requisite: Nil

Recommended Texts

1. Kenneth Cukier and Viktor Mayer-Schönberger (2014). Big data: *a revolution that will transform how we live, work, and think*. London: Eamon Dolan/Mariner Books
2. Jimmy Lin and Chris Diyer (2010). *Data-intensive text processing with mapreduce*. California: Morgan & Claypool Publishers

Suggested Readings

1. Andy Konwinski, Holden Karau, Matei Zaharia, and Patrick Wendell (2015). *Learning spark: lightning-fast big data analysis*. California: O'Reilly Media.
2. Anand Rajaraman and Jeff Ullman (2011). *Mining of massive datasets*. Cambridge: Cambridge Press.

BS
SOFTWARE ENGINEERING

CMPC-5201 **Programming Fundamentals** **4 (3+1)**

In this course the students will gain a broad understanding of the field of computing and fundamentals of computer programming. The student will acquire introductory skills in problem analysis, solution design, and program construction. Through practical programming activities, the student will gain an appreciation of the nature and history of computer programming. Students will learn to use pseudo-code and visual modeling to prepare clear and accurate program documentation and models. Also, they will examine working programs to identify their structures which will help them to use appropriate structures in their designed algorithms. At the end of the course students will be able to get a fundamental understanding of software development methodologies, including modular design, pseudo code, flowcharting, structure charts, data types, control structures, functions, arrays and the mechanics of running, testing, and debugging. Students will develop a project in the course that utilize logical algorithms from specifications and requirements statements.

Contents

1. Overview of Computer Programming
2. Principles of Structured and Modular Programming
3. Overview of Structured Programming Languages
4. Algorithms and Problem Solving
5. Program Development: Analyzing Problem, Designing Algorithm/Solution
6. Fundamental Programming Constructs, Data Types
7. Basics of Input and Output
8. Unary and Binary operators. Arithmetic (Expression) in C.
9. Selection and Decision (If, If-Else, Nested If-Else, Switch Statement and Condition Operator)
10. Repetition (While and For Loop, Do-While Loops)
11. Break Statement, Continue Statement
12. Pointers: Pointer expressions, Pointers and arrays, Pointers in functions.
13. Pointers: Static and dynamic memory allocation, Memory Management using Pointers.
14. Control Structures, Functions, Arrays, Records
15. Files (Input-Output), Testing & Debugging.

Pre-requisites: Nil

Recommended Texts

1. Paul Deitel and Harvey Deitel. (2012). *C how to program,* (7th ed.). New York: Prentice Hall

Suggested Readings

3. Stephen G. Kochan. (2013). *Programming in c*, (4th ed.). New York: Addison-Wesley Professional.
4. Deitel, P., & Deitel, H. (2011). *Java how to program,* (9th ed.). New York: Prentice Hall.

CMPC-5202 **Object Oriented Programming** **4 (3+1)**

This course provides an introduction to object-oriented programming (OOP) using the Java/C++ programming language to students with a procedural background. Its main objective is to teach the basic concepts and techniques which form the object-oriented programming paradigm. Other objectives of the course include becoming familiar with breaking down a problem into objects rather than procedures and learn object-oriented programming in C++. Students will learn to Identify the objects & their relationships to build object-oriented solution. Students will perform object-oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O. and other standard language constructs. The course briefly covers the mapping of UML design to Java/C++ implementation and object-oriented considerations for software design and reuse. The course also relates Java/C++ to GUI, databases, and real-time programming. At the end of the course students may able to understand concepts like classes, inheritance, polymorphism and abstraction etc.

Contents

1. Objects and Classes, Abstraction, Encapsulation.
2. Final Classes, Nested and Inner Classes.
3. Inheritance, Abstract Classes, Concrete Classes, Inheritance and Encapsulation.
4. The is-a Relationship, Inheritance via Abstract Classes, Extending the Hierarchy, Up casting and Down casting, Interfaces.
5. Composition, the has-a Relationship.
6. Polymorphism.
7. Polymorphism, Dynamic (or Late) Binding.
8. Interfaces and Polymorphism.
9. The Wrapper Classes, Boxing and Un-Boxing, Packages.
10. Exceptions and Exception Handling.
11. File Systems and Paths, File and Directory Handling and Manipulation, Input/output Streams, Reading Binary Data, writing binary Data, Writing Text (Characters), Reading Text (Characters), Logging with Print Stream, Random Access Files, Object Serialization.
12. Collections, for-each Loop.
13. GUI Concepts, Components and Containers, Abstract Windows Toolkit and Swing,
14. Windows and Frames, Layout Managers, Panels.
15. Event-Driven Programming, The delegation Event Model.
16. Event Classes, Mouse Events, Keyboard Events, Using Actions.
17. Component and J Component, Buttons, Labels, Text Fields, Text Areas, Dialog Boxes, Checkboxes and Radio Buttons, Menus, J-Slider, J-Tabbed Pane.

Pre-requisites: Programming Fundamentals

Recommended Texts

1. Deitel, P., & Deitel, H. (2016). *C++ how to program,* (10th ed.). Harvey: Pearson.
2. Robert Lafore. (2001) *Object oriented programming in c++,* (3rd ed.). New York: SAMS.

Suggested Readings

1. Gaddis, T., & Sengupta, P. (2012). *Starting out with c++: from control structures through objects,* (9th ed.). New Dehli: Pearson.
2. Deitel, P., & Deitel, H. (2011). *Java how to program,* (9th ed.). New York: Prentice Hall.

CMPC-5104 **Discrete Structure** **3 (3+0)**

The course provides a solid theoretical foundation of discrete structures as they apply to computer science problems and structures. Objectives of the course includes to obtain a useful mastery of discrete structures and methods basic to further work in computer science. Also, the ability to formulate and solve applied problems, to analyze and interpret algorithms and functions and to use them effectively. Students will learn how to use mathematical notation and solve problems using mathematical tools. At the end of the course students will develop the mathematical foundations necessary for more specialized subjects in computer science, including data structures, algorithms, and compiler design. In this course students will discover that logical propositions are the underlying model of discrete systems. From this they will develop algorithms and prove their efficacy. Topics include propositional and predicate logic, basic proof techniques, set algebra and Boolean algebra, recursion and induction, trees and graphs, introductory combinatorics, and matrix algebra with an emphasis on applications in computer science.

Contents

1. Logic: Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Methods of Proof.
2. Sets & Functions, Sequences and Summations.
3. Algorithms: The Growth of Functions, Complexity of Algorithms, the Integers and Division, Matrices.
4. Number Theory and Cryptography.
5. Advanced Counting Techniques: Recurrence Relations, Solving Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion & its Application.
6. Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.
7. Graph: Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.
8. Trees: Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

Pre-requisites: Nil

Recommended texts

1. Rosen, K. H., & Krithivasan, K. (2012). *Discrete mathematics and its applications: with combinatorics and graph theory*, (7th ed.). New York: Tata McGraw-Hill Education.
2. R. Johnsonbaugh. (2015). *Discrete mathematics*, (7th ed.). Chicago: Pearson.

Suggested Readings

1. S. B. Maurer and A. Ralston. (2005). *Discrete algorithmic mathematics* (3rd ed.). New York: A K Peters/CRC Press.
2. Kolman, R. Busby and S. C. Ross. (2018). *Discrete mathematical structures* (6th ed.). New York: Prentice Hall.

CMPC-5105 **Software Engineering** **3 (3+0)**

At the end of the course the students will be able to explain how various software development models and software development life cycles are applied. Presents the fundamentals concepts of project management. Important Requirements modeling, fact-finding techniques. Have knowledge of software configuration management. Students will then learn about the Software Development Life Cycle (SDLC) followed by software modeling using Unified Modeling Language (UML), a standardized general-purpose modeling language used to create visual models of object-oriented software. At the end of the course students may able to learn about five major phases of the SDLC: requirements gathering, requirements analysis, design, coding/implementation, and testing. You will also learn about project management for the purpose of delivering high-quality software that satisfies customer needs and is within budget. There is a focus on software testing, from unit testing to the testing of software releases. Project management and professional software engineering practice will also be covered. Case studies provide practical examples for many of these concepts.

Contents

1. The Nature of Software, Nature of Web Apps, the Software Process, Software Engineering Practice.
2. Generic Process Models. Specialized Process Models. Systems Analysis and Design.
3. Business Information Systems.
4. Introduction to SDLC, SDLC Phases, System Planning, Preliminary Investigation, SWOT Analysis.
5. Strategic Planning. Information Systems Projects. Requirements Engineering. Data & Process Modeling.
6. Design within the Context of Software Engineering. Design Models.
7. System Architecture. Architectural Styles.
8. User Interface Design.
9. Software Quality Assurance.
10. Validation Testing, System Testing, Internal and External View of Testing.
11. Project Management Concepts.
12. Project Scheduling.
13. Risk Management. Maintenance and Reengineering.

Recommended Texts

1. Roger S. Pressman, (2007). *Software Engineering: A Practitioner's Approach (*7th ed.). New York: McGraw-Hill
2. Gary B. Shelly, Thomas J. Cashman and Harry J. Rosenblatt, (2011). *Systems Analysis and Design (*7th ed.). Boston: Cengage Learning Course Technology

Suggested Readings

1. Ian Sommerville, Addison Wesley, (2018). *Software engineering* (8th ed.). New York: Pearson.
2. Gary B. Shelly, *Thomas J. Cashman and Harry J. Rosenblatt*, (2011). *Systems analysis and design (*7th ed.). Boston: Course Technology.

CMPC-5203 **Data Structures and Algorithms** **4 (3+1)**

This course provides in-depth coverage theory, practice and methods of data structures and algorithm design. This course provides an understanding of the layered approach that makes design, implementation and operation and identify aspect of complex operating system. Students will be able to assess how the choice of data structures and algorithm design methods impacts the performance of programs. Students may able to get a clear understanding of how to choose the appropriate data structure and algorithm design method for a specified application. Write programs using object-oriented design principles. Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, tournament trees, binary search trees, and graphs and writing programs for these solutions. Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound and writing programs for these solutions. At the end of the course, students will be assigned a semester project.

Contents

1. Introduction to Course, Basic Object Orientation concepts, Properties of Algorithm,
2. ADT, Basic Operations, Reading, Writing, Insertion, Deletion, Merging, Binary Search.
3. Introduction to Sorting types and Techniques, Logical and Algorithmic Implementation of Bubble, Selection Sort, Insertion, Quick Sort, Merge Sort.
4. The Stack ADT, Expressions, Postfix Notation, Infix to postfix, postfix evaluation Introduction to Recursion, Examples of Recursion, Writing Recursive Programs
5. The Queue ADT, Variation of Queue ADT i.e. Circular and Double Ended Queue
6. Priority Queues, Introduction to Pointers, Linear single Link
7. Linked Stacks and Queues, Linear Doubly Linked list
8. Circular Lists: Implementation of queues and stacks, Doubly Link List
9. Introduction to Trees, Tree Terminology, Logical construction and Representation of Trees
10. Binary Search Tree, Implementation and Applications of BSTs
11. Heaps and Heaps as Priority Queues, Introduction to Balanced and AVL Trees.
12. Hashing, Overflow Handling, Open Addressing, Chaining
13. Elementary Graph Operations, DFS, BFS, Spanning Trees
14. Shortest path algorithms: Dijkstra Algorithm, Minimum Cost Spanning Trees.

Pre-requisites: Object Oriented Programming

Recommended Texts

1. Michael T. Goodrich, Roberto Tamassia, and David Mount. (2011). *Data structures and algorithms in c++*, (2nd ed.). New York: John Wiley & Sons.
2. Adam Drozdek. (2012). *Data structures and algorithms in c++* (4th ed.). New York: Cengage Learning.

Suggested Readings

1. Carrano, F. M., & Savitch, W. J. (2003). Data structures and abstractions with java. New York: Prentice Hall.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. (2009). Introduction to algorithms, (2nd ed.). New York.

CMPC-5204 **Operating System** **4 (3+1)**

After completing the course, the students will be able to understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems. Analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions. Demonstrate the knowledge in applying system software and tools available in modern operating systems. Topics to be discussed include a brief history of OS's and their design and development. The course will cover major components and the algorithms and implementation techniques used to create them. At the end of the course students may have a solid understanding of the theoretical, the operational, and the implementation underpinnings of the modern computing infrastructure to be able to effectively utilize the whole spectrum of the modern computing infrastructure, including computer hardware, software, programming environments, operating systems, and networking environments.

Contents

1. Operating-System Structure, Operating-System Operations
2. Process management, Memory Management, Storage Management
3. Protection and Security, Protection and Security, Distributed Systems
4. Special-Purpose Systems, Computing Environments, Operating-System Services
5. User Operating-System Interface, Virtual Machines, System Calls, Operating-System Generation, Types of System Calls, System Boot, System Programs
6. Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, Communication in Client- Server Systems. Threads: Multithreading Models, Thread Libraries
7. Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Thread Scheduling, Algorithm Evaluation, Monitors, The Critical-Section Problem, Peterson's Solution
8. Classic Problems of Synchronization, System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection
9. Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation
10. Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, File-System Implementation: File-System Structure,
11. Log-Structured File Systems, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery,
12. The Security Problem, Computer-Security, Program Threats, Classifications
13. User Authentication, Implementing Security Defenses, Firewalling to Protect Systems.

Pre-requisites: Nil

Recommended Texts

1. Silberschatz, A., Peterson, J. L., & Galvin, P. B. (1991). *Operating system concepts* (9^{th} ed.). New York: John Wiley & Sons.
2. Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4^{th} ed.). New York: Pearson/Prentice Hall.

Suggested Readings

1. Stallings, W. (2009). *Operating systems: internals and design principles*. New York: Pearson/Prentice Hall.

CMPC-5205 **Database System** **4 (3+1)**

The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS concepts. Identify functional dependencies and resolve database anomalies by normalizing database tables. Use Structured Query Language (SQL) for database definition and manipulation in any DBMS. Students may able to learn different type of data models, normalization and SQL language. Several programming projects are assigned in the course involving the use of a database management system. Objectives include basic understanding of Database systems and their working as well as use. They will also learn Design conceptual, logical and physical database schemas using different data models. They will use Structured Query Language (SQL) for database definition and manipulation in any DBMS. Topics include data models (ER, relational, and others); query languages (relational algebra, SQL, and others); implementation techniques of database management systems.

Contents

1. Databases Overview: Basics and fundamentals
2. Data Models, Advantages of DB, Cost and Risk, Components of the DB Environment, The Database Development Process
3. E-R Model, Modelling Rules, Modelling Entities, Attributes and relations
4. Enhanced E-R Model, Subtype/Supertypes, Specifying Constraints in Relationships
5. Relational Data Model, Integrity Constraints, Transforming EER Diagrams into Relations
6. Introduction to Normalization, first, second and third normal form.
7. Designing Physical DB, Designing Fields: Data Types, Coding Techniques, Handling Missing Data
8. Deformalizing and Partitioning Data, File Organizations
9. Introduction to SQL, The SQL Environment, Defining a Database in SQL
10. Basic SQL Commands for Processing Single Tables
11. SQL Commands for Processing Multiple Tables using Joins and Subqueries
12. Client/Server Architectures, Databases in a Two-Tier and Three-Tier Architectures
13. Web Application Components, Databases in Three-Tier Applications
14. The Roles of Data and Database Administrators, Database Backup and Recovery, Types of Database Failure, Disaster Recovery
15. Concurrent Access, Serializability, Locking Mechanisms, Dictionaries and Repositories
16. Lab work should be carried out to develop students' Database Skills

Recommended Texts

1. Hoffer, J. A., Ramesh, V., & Topi, H. (2011). *Modern database management*. New York: Prentice Hall.
2. Connolly, T. M., & Begg, C. E. (2005). *Database systems: a practical approach to design, implementation, and management*. Boston: Addison-Wesley

Suggested Readings

1. Elmasri, R., & Navathe, S. (2017). *Fundamentals of database systems* (Vol. 7). New Delhi: Pearson.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (1997). *Database system concepts* (Vol. 5). New

York: McGraw-Hill.

CMPC-6201 **Computer Networks** **4 (3+1)**

This course provides an introduction to key concepts related to computer communication. Describe the key terminologies and technologies of computer networks. Explain the services and functions provided by each layer in the Internet protocol stack. Identify various internetworking devices and protocols, and their functions in a network. Analyze working and performance of key technologies, algorithms and protocols. Build Computer Network on various Topologies. It considers how to design networks and protocols for diverse situations, analyses several application and support protocols from a distributed systems viewpoint, and identifies significant problem areas in networked communications. Upon successful completion, students will have the knowledge and skills to describe, analyze and evaluate various related technical, administrative and social aspects of specific computer network protocols from standards documents and other primary materials found through research. Students will also get a hands-on experience of a packet tracer tool and will learn how to make basic networks and topologies.

Contents

1. Introduction to networks and protocols architecture.
2. Basic concepts of networking, network topologies and the Internet.
3. Layered architecture and the OSI model.
4. Physical layer functionality, data link layer functionality and the TCP/IP protocol architecture.
5. Multiple access techniques, WAN Technologies and protocols, circuit switching and packet switching.
6. Wireless networks, Cellular Network Generations and LTE-Advanced.
7. LAN technologies, LAN protocol architecture and virtual LANs
8. MAC addressing.
9. Networking devices, bridges, hubs and switches
10. Network layer protocols, Principles of Internetworking, IPv4 and IPv6.
11. IP addressing, Internet Protocol Operation, virtual private networks and IP Security and Subletting, CIDR.
12. Transport layer protocols, ports and sockets and connection-oriented transport protocol mechanisms.
13. Routing protocols OSPF, EIGRP, RIP and routing in packet-switching networks
14. Connection establishment, flow and congestion control, effects of congestion, TCP congestion control and datagram congestion control protocol.
15. Application layer protocols, electronic mail (SMTP and MIME), Web Access: HTTP and DNS.
16. Latest trends in computer networks, real-time traffic and voice over IP.

Recommended Texts

1. Stallings, W. (2007). *Data and computer communications*. New Delhi: Pearson Education.

Suggested Readings

1. Kurose, J. F. (2005). *Computer networking: A top-down approach featuring the internet, 3/E*. New Delhi: Pearson Education.
2. Tannenbaum, A. S. (2002). *Computer networks*. New Delhi: Pearson Education.
3. Forouzan Behrouz, E. (2003). *Data communication & computer networks*. New Delhi: Tata

McGraw Hill Publications.

CMPC-6101 **Information Security** **3 (3+0)**

At the end of the course, the students will be able to explain key concepts of information security. They will be able to discuss legal, ethical, and professional issues in information security. They can apply various security and risk management tools for achieving information security and privacy. Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications. Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath. Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges. Determine appropriate mechanisms for protecting information systems ranging from operating systems to database management systems and to applications. It will be easy to identify appropriate techniques to tackle and solve problems in the discipline of information security.

Contents

1. Information security foundations, security design principles; security mechanisms.
2. Vulnerabilities and protections: Malwares.
3. Hash functions.
4. Digital signatures.
5. Key management.
6. Authentication: Access control.
7. Symmetric cryptography: Symmetric Encryption, DES.
8. AES, Cipher Block Modes.
9. Asymmetric cryptography: HMAC, The RSA, Diffie- Hellman
10. Database security: The need, DBMS Relational Databases, SQL Injection Attacks, Database Access Control, Inference, Database Encryption
11. Network security: Secure E-Mail and S/MIME, DomainKeys Identified Mail, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, Wireless Security, Mobile Device Security
12. Firewalls: The Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls
13. Intrusion detection: Intruders, Intrusion Detection, Analysis Approaches, Types of IDS.
14. Security policies, policy formation and enforcement, risk assessment.
15. Cybercrime, law and ethics in information security, privacy and anonymity of data

Recommended Books

1. Stallings, W., Brown, L., Bauer, M. D., & Bhattacharjee, A. K. (2012). *Computer security: principles and practice*. New York: Pearson Education.
2. Whitman, M. E., & Mattord, H. J. (2011). *Principles of information security*. Boston: Cengage Learning.

Suggested Readings

1. Gollmann, D. (2011). *Computer security*. The Atrium: Wiley & Sons.
2. Easttom, W. (2013). *Computer security fundamentals with myiTcertificationlab bundle*. New York: Pearson IT Certification

CMPC-6701 **Capstone I** **3(0 +3)**

The Capstone Project is an opportunity for participants to put their leadership competencies into practice. Capstone Project allows the student to identify and develop a project that puts into practice the leadership skills and competencies learned during the courses of study. In Capstone Project I students are working in a group form and finalize the Project Proposal. On approval of Project proposal from Project Coordinator group has to work on finalizing the Software Requirement Specifications. In this phase groups are utilizing the Software Engineering guidelines to finalize the Software Requirement Specifications. On finalization of Requirement documentation software Design process is initiated. In this guideline of software design specifications are followed. By developing Project potters and videos groups can improve and sharpen their multimedia skills. Posters and Videos competitions are held regularly to improve competition among project groups. At the end each group has to give presentation and explain in detail all the work completed during semester.

Contents:

1. Identifying Project Area
2. Filtering and Finalizing Project
3. Project Proposal
4. Identifying Project Scope
5. Finalizing project Description
6. Non-Functional Requirements
7. Functional Requirements
8. Identifying Project High level Plans
9. Overall project Description
10. System Architecture
11. Literature Review (In case of Research base project)
12. Software Cost, Time and Effort Estimations
13. Identifying main use cases.
14. Finalizing Software Development tool
15. Project Poster Development
16. Project Video Development

Pre-Requisite: Databases System, Software Engineering.

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction*. Switzerland: Springer International Publishing.

2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addision-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc GrawHill
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York:

Pearson.

CMPC-6702 **Capstone II** **3(0+3)**

Capstone Project II is a continuation of Capstone Project I. This course provides students with the opportunity to apply the knowledge and skills acquired in their courses to a specific problem or issue. To allow students to extend their academic experience into areas of personal interest, working with new ideas, issues, organizations, and individuals. Main focus of this course is development of Capstone Project I and according to details and area finalized. Group are developing functional requirements and collaborate with the project supervisors and coordinators throughout the semester. At the end of project groups has to present running project to the Project evaluation Committee. In Capstone Project II student can consult with the supervisor or expert from industry. Capstone Project II engages the students with real world market-oriented problems and complexities. Groups has to put in extra effort along with final semester courses to cover the complexities and deadlines of Capstone Project II.

Contents

1. System Sequence Diagram,
2. Domain Model
3. State Chart Diagrams and Implementation
4. Modeling Generalization, Design Class Diagram, Mapping Model to Domain Model
5. Implementation of Design Class Diagram, Coding patterns, Mapping Design to Code
6. Implementation according to Project Area, Web base application, Android/IOS Development, Research base project, Network base project development, Cloud based project, Hardware supported, IOT based, AI and Machine Learning based projects.
7. Implementation design according to project development area.
8. For Database oriented projects details in points 9-10 are followed.
9. SQL Commands for Processing Multiple Tables using Joins and Sub queries.
10. Client/Server Architectures, Databases in a Two-Tier Architecture, Three-Tier Architectures.
11. Web Application Components, Databases in Three-Tier Applications.
12. For Network/cloud based/AI and Machine Learning based projects implementation
13. Project framework development according to specific area.
14. Applying Software Quality Assurance, Software Testing Strategies, Strategic Issues, Test.
15. Validation, System and Integration Testing.

Pre-Requisite: Required Programming Courses

Recommended Texts

1. Weyers, B., Bowen, J. (2017). *The handbook of formal methods in human-computer interaction*. Switzerland: Springer International Publishing.

2. Booch, G., Maksimchuk, R. (2008). *Object-oriented analysis and design with applications*. Boston: Addision-Wesley.

Suggested Readings

1. Pressman, R. S. (2007). *A practitioner's approach. software engineering*. New York: Mc GrawHill
2. Hoffer, J. A., Ramesh, V., & Topi, H. (2016). *Modern database management*. New York: Pearson.

SECC-5201 Software Requirement Engineering 3(3+0)

At the end of the course the students will be able to understand the Issues in Requirements Engineering. Also, will be able to understand and apply Requirements Engineering Process. To use the requirements Elicitation and specification. To use Formal Techniques and to understand modeling and analysis of Non-Functional Requirements. Manage and control changes in Requirement. Objectives of the course is to help students gain understanding of what is requirement engineering, the different stages and the concepts of requirement engineering. To describe in detail the different activities of requirement engineering. To give students practical knowledge on how to make certain requirement engineering artifacts. At the end of this course students may able to model software requirements rigorously according to the latest requirements engineering standards. They will also be able to generate and maintain a software requirements specification document. Other objectives of course include to conduct a prioritization process for software requirements according to different approaches.

Contents

1. Essential Software Requirement, Bad Requirements, Characteristics and Benefits Requirement Engineering, Requirements from the Customer's Perspective.
2. Requirement Engineering Processes.
3. The Requirements Analyst Role, Defining the Vision through Business Requirements.
4. Dealing with Customers: Elicitation Techniques, Interviews, Surveys, Workshops, Classifying Customer Input, Incomplete Requirements (Finding Missing Requirements).
5. Prototype Categories, evaluation, Risks, Validating the Requirements: Requirements Review and Inspection, Requirements Review Challenges, Acceptance Criteria.
6. The Software Requirements Specification, Labeling, Dealing with Incompleteness, User Interfaces and the SRS, A Software Requirements Specification Template, Data Dictionary.
7. Non-Functional Requirements: Non-functional Requirements Using P language.
8. Setting Requirements Priorities: Why Prioritize Requirements? Play with Priorities.
9. Special Requirements Challenges.
10. Requirements Development Plans, Estimation, Scheduling, Code, Tests and Success.
11. Requirements Management Principles and Practices.
12. Requirements Creeping, Managing Scope Creep, The Change Control Process, The Change Control Board, Change-Control Tools, Measuring Change Activity, Impact Analysis.
13. The Requirements Traceability Matrix, Tools for Requirements Traceability.
14. Tools for Requirements Management.

Pre-requisite: Discrete Structure

Recommended Texts

1. Karl Wiegers. (2013). *Software requirements* (3rd ed.). Redmond: Microsoft Press.
2. Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: processes and techniques*. New York: Wiley Publishing.

Suggested Readings

1. Thayer, R. H., Bailin, S. C., & Dorfman, M. (1997). *Software requirements engineerings*. Washington: IEEE Computer Society Press.

SECC-5202 Human Computer Interaction 3(3+0)

This course will equip the students with the tools and techniques required to design user-friendly interactive systems, latest theories, principles guidelines in Human-Computer Interaction (HCI), User Experience (UX), usability and Interaction Design, Moreover the students will be able to understand the entire user-centered design and evaluation process; from understanding user needs to design interactive systems that meet those needs, to evaluate the usability of those (and existing systems) through user research. The students will be able to explain the capabilities of both humans and computers from the viewpoint of human information processing. Students will also have an idea about understanding users' problems, designing usable user interfaces and evaluating these user interfaces and UX with end users. Through a set of exercises, students will acquire the ability to critique problems that exist in current interactive software and websites. At the end of the course students may be able to learn several usability evaluation methods.

Contents

1. The human: Input–output channels, Human memory, Cognition.
2. The interaction, Frameworks and HCI, Ergonomics, Interaction styles, Elements of the WIMP
3. Interaction design basics, Interaction types, User focus, Scenarios, Navigation design, Screen design and layout, Iteration and prototyping. considering assistive technologies and culture
4. User and task analysis, HTA, uses of task analysis, accessibility, standards, User experience design, and cognitive psychology, anthropology and ethnography, and ergonomics/human factors.
5. Design rules, Standards, benchmarks, Guidelines, Golden rules and heuristics, HCI patterns.
6. Evaluation techniques and Goals, Evaluation through expert analysis for evaluation of user-centered design, articulate evaluation criteria and compliance to relevant standards, Metrics and Measures for Evaluation
7. Universal design principles, Multi-modal interaction, designing for diversity.
8. User support, approaches to user support, designing user support systems.
9. Groupware systems, Computer-mediated communication, Meeting and decision support systems, Shared applications and artifacts, Frameworks for groupware.
10. Ubiquitous computing and augmented realities, Virtual and augmented reality, Information and data visualization.
11. Design an interactive application, applying a user-centered design cycle and related tools and techniques (e.g., prototyping
12. Usability inspection methods, Usability testing methods, New Interaction Technologie

Pre-Requisite: Software Engineering

Recommended Texts

1. Preece, J., Rogers, Y., & Sharp, H. (2019). *Interaction design: beyond human-computer interaction* (5th ed.). New York: John Wiley & Sons Ltd.
2. Dix, A., Finlay, J., Abowd, G.D., & Beale, R. (2004*). Human computer interaction.* (3rd ed.). New Delhi: Prentice Hall.

Suggested Readings

1. Ben Shneiderman and Catherine Plaisant. (2016) *Designing the User Interface: Strategies for effective human-computer interaction* (6th ed.). New York: Pearson Inc,
2. Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). *About face: the essentials of interaction design*. (4th ed.). New York: John Wiley & Sons

SECC-5203 Software Design & Architecture 3(2+1)

At the end of the course the students will be able to understand the role of design and its major activities within the OO software development process, with focus on the Unified process. Students may able to comprehend the advantages of consistent and reliable software design. Design OOD models and refine them to reflect implementation details. Students can learn to apply and use UML to visualize and document the design of software systems. Implement the design model using an object-oriented programming language. In this course students will also study the ways in which software architecture (subroutines, classes, functions) are represented, both in UML and other visual tools. At the end of the course students will also learn the common architectures, their qualities, and tradeoffs. How architectures are evaluated, what makes a good architecture, and an architecture can be improved. And how the architecture touches on the process of software development.

Contents

1. Software Architecture Introduction.
2. Characteristics of SA, Importance of SA, SA Business Cycle and Software Processes, SA History, "Good" Architecture, Architectural Patterns, Reference Models, and Reference Architectures, Architectural Structures and Views
3. SA Case Study: A-7E Avionics System, Business Cycle, Requirements and Qualities
4. Creating Architecture: Understanding Quality Attributes, Functionality and Architecture.
5. Achieving Qualities.
6. Global Analysis
7. Conceptual Architecture View.
8. Module Architecture View.
9. Execution Architecture View, Code Architecture View.
10. Designing & Documenting the Architecture.
11. Analyzing Architectures: The ATAM - A Comprehensive Method for Architecture Evaluation, Participants in the ATAM, Outputs of the ATAM, Phases of the ATAM, The Nightingale System: A Case Study in Applying the ATAM
12. The CBAM: A Quantitative Approach to Architecture Design Decision Making
13. Reconstructing Software Architectures: Introduction, Information Extraction, Database Construction, View Fusion, Reconstruction

Pre-requisite: Nil

Recommended Texts

1. Hofmeister, C., Nord, R., & Soni, D. (2000). *Applied software architecture*. Ontario: Addison-Wesley Professional.
2. Bass, L., Clements, P., & Kazman, R. (2015). *Software architecture in practice*. New York: Addison-Wesley Professional

Suggested Readings

1. Qian, K., Fu, X., Tao, L., & Xu, C. W. (2010). *Software architecture and design illuminated*. New York: Jones & Bartlett Learning.
2. Taylor, R. N., Medvidovic, N., & Dashofy, E. (2009). *Software architecture: foundations, theory, and practice*. New York: John Wiley & Sons.

SECC-6201 **Software Construction & Development** **3(3+0)**

Software Construction introduces fundamental principles and techniques of software development, i.e., how to write software that is safe from bugs, easy to understand, and ready for change. At the end of the course the students will be able to apply a wide variety of software construction techniques and tools, including state-based and table-driven approaches to low-level design of software. Objectives of the course includes to understand and apply collaborative construction, understanding refactoring & its strategies and to understand layout and styling of developing software. To demonstrate by examples the key construction life cycle models. Since construction is a hands-on intensive endeavor, students are challenged with hands-on exercises that mimic real-world software construction challenges. To explain the application of software construction tools such as GUI builders, unit testing tools, profiling, performance analysis and slicing tools. At the end of the course students may able to understand the concepts like software quality landscape, documenting code, integration techniques etc.

Contents

1. Software Construction: What Is Software Construction? Why Is Software Construction Important? Metaphors for Software Development, The Importance of Metaphors.
2. Prerequisites: Importance of Prerequisites, Type of Target Software, Problem-Definition Prerequisite, Requirements Prerequisite, Architecture Prerequisite, Time Constraints.
3. Key Construction Decisions: Choice of Programming Language, Programming Conventions, Localization Aspects of Technology, Selection of Construction Practices.
4. Design in Software Construction: Design Challenges, Key Design Concepts.
5. Defensive Programming.
6. The Software-Quality Landscape: Characteristics of Software Quality, Techniques for Improving Software Quality, Relative Effectiveness of Quality Techniques.
7. Collaborative Construction., Refactoring, Program Size & Software Construction.
8. Managing Construction: Encouraging Good Coding, Configuration Management.
9. Integration: Importance of the Integration Approach, Integration Frequency—Phased or Incremental? Incremental Integration Strategies, Daily Build and Smoke Test.
10. Programming Tools: Design Tools: Source-Code Tools, Executable-Code Tools, Tool-Oriented Environments, Building Your Own Programming Tools, Tool Fantasyland.
11. Layout and Style: Layout Fundamentals, Layout Techniques, Layout Styles, Laying Out Control Structures, Laying Out Individual Statements, Laying Out Comments, Laying Out Routines
12. Self-Documenting Code: External Documentation, Programming Style as Documentation, To Comment or Not to Comment, Keys to Effective Comments.

Pre-requisite: Nil

Recommended Texts

1. Wirth, N., Wirth, N., Wirth, N., Informaticien, S., & Wirth, N. (1996). *Compiler construction* (Vol. 1). New York: Addison-Wesley.

Suggested Readings

1. Meyer, B. (1988). Object-oriented software construction. New York: Prentice hall.
2. Alagic, S., & Arbib, M. A. (2013). The design of well-structured and correct programs. New York: Springer Science & Business Media.

SECC-6101 Software Quality Engineering 3(3+0)

Software Construction introduces fundamental principles and techniques of software development, i.e., how to write software that is safe from bugs, easy to understand, and ready for change. At the end of the course the students will be able to apply a wide variety of software construction techniques and tools, including state-based and table-driven approaches to low-level design of software. Objectives of the course includes to understand and apply collaborative construction, understanding refactoring & its strategies and to understand layout and styling of developing software. To demonstrate by examples the key construction life cycle models. Since construction is a hands-on intensive endeavor, students are challenged with hands-on exercises that mimic real-world software construction challenges. To explain the application of software construction tools such as GUI builders, unit testing tools, profiling, performance analysis and slicing tools. At the end of the course students may able to understand the concepts like software quality landscape, documenting code, integration techniques etc.

Contents

1. A Quality Principles, Benefits of Quality, Organization and Process Benchmarking Ethical and Legal Compliance: ASQ Code of Ethics, Legal and Regularity Issues
2. Standards and Models: ISO 9000 Standards, IEEE Software Engineering Standards, SEI Compatibility Maturity Model Integration (CMMI).
3. Leadership Skills: Organizational Leadership, Facilitation Skills, Communication Skills.
4. Team Skills, Team Management, Team Tools
5. Quality Management System: Quality Goals and Objectives, Customers and Other Stakeholders
6. Methodologies for Quality Management: Cost of Quality, Process Improvement Models
7. Audit Types, Audit Roles and Responsibilities, Audit Process.
8. Project Tracking and Control, Tracking Methods, Project Reviews and Program Reviews
9. Software Verification and Validation Method, Software Product Evaluation,
10. Testing Planning and Design.
11. Reviews and Inspection: Peer Reviews, Formal Versus Informal Reviews, Types of Peer Reviews, Walk-Throughs, Inspections, Technical Reviews.
12. Test Execution Documentation: Test Execution, Test Case, Test Procedure, Test Log
13. Customer Deliverables: Peer Reviews, Development Testing, Development Audits, Pilots, Installation Testing, Customer/User Testing.
14. Configuration Control and Status Accounting: Item, Baseline, Version Control, CCB, Concurrent Development, and Status Accounting, Configuration Audit

Pre-requisite: Discrete Structure

Recommended Texts

1. Westfall, L. (2009). *The certified software quality engineer handbook*. Nebraska: American Society for Quality Press.
2. Godbole, N. S. (2004). *Software quality assurance: principles and practice*. Oxford: Alpha Science Int'l Ltd.

Suggested Readings

1. McCaffrey, J. D. (2009). Software testing: fundamental principles and essential knowledge. New York: BookSurge Publishing. .

SECC-6102 **Web Engineering** **3(3+0)**

Web Engineering introduces a structured methodology utilized in software engineering to Web development projects. The course addresses the concepts, methods, technologies, and techniques of developing Web sites that collect, organize and expose information resources. This course introduces students to the discipline of web engineering including the methods and techniques used in web-based system development. At the end of the course the students will be able to discuss how web standards impact software development. Describe the constraints that the web puts on developers. Design and Implement a simple web application. Review an existing web application against a current web standard. At the end of the course students will be able to understand the concepts of web applications modeling and architecture, overall web application development process, security of web applications and semantic web. Students may also be able to learn and use some of the client-side and server-side languages used to manipulate information on the World Wide Web.

Content

1. An Introduction to Web Engineering.
2. Requirements Engineering for Web Applications.
3. Web Applications Modeling.
4. Web Application Architectures.
5. Technology-away Web Application Design: Web Design from an Evolutionary Perspective, Presentation Design, Interaction Design, Functional Design, Outlook.
6. Technologies for Web Applications: Fundamentals, Client/Server Communication on the Web, Client-side Technologies, Document-specific Technologies, Server-side Technologies.
7. Testing Web Applications.
8. Operation and Maintenance of Web Application.
9. Web Project Management.
10. Web Application Development Process, Usability of Web Applications.
11. Performance of Web Applications, Security for Web Applications.
12. The Semantic Web: Fundamentals, Technological Concepts, Specifics of Semantics Web Applications, Tools.

Pre-requisite: Nil

Recommended Texts

1. Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2006). *Web engineering*. New York: Wiley.
2. Chopra, R. (2016). *Web engineering*. New Delhi: Prentice-Hall.

Suggested Texts

1. Mendes, E., & Mosley, N. (2006). *Web engineering*. Springer Science & Business Media.
2. Roger S. (2008). *Web engineering: a practitioners' approach*. New York: Pressman, McGraw Hill.
3. Goodman, D. (2002). *Dynamic html: the definitive reference: a comprehensive resource for html, css, dom & javascript*. California: O'Reilly Media, Inc.

SECC-6103 Software Project Management 3(3+0)

The course provides an in-depth examination of project management principles and modern software project management practices. At the end of the course the students will be able to explain principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding. Critically evaluate and discuss the issues around project management and its application in the real world with course participants and learners. Objectives of the course includes: Understanding the five process groups and nine knowledge areas of the PMI BOK. To understand approaches for managing and optimizing the software development process. To understand efficient techniques for managing each phase of the systems development lifecycle and to use and application of tools to facilitate the software project management process (e.g. Microsoft Project). Methods for managing and optimizing the software development process are discussed along with techniques for performing each phase of the systems development lifecycle.

Content

1. Introduction: Software Project Versus Other Type of Projects
2. Introduction of PM Tools, PMI 's Knowledge Areas, Technical Fundamentals in SPM, Lifecycle Relationships, Classic Mistakes Product-Process-Peoples-Technology Mistakes
3. PMI Framework, PMI Process Groups: Process Initiating Process Group, Planning Process Group, Executing Process, Process Monitoring and controlling, Closing Process Group, Project Charter, Statement of Work.
4. Understanding Organizations, Organizational Structures, Functional -Project –Matrix Organizational Impact on Projects, identifying stakeholders: Define Responsibilities, Authority Relationships, Position Qualifications.
5. Project Planning.
6. Project Evaluation, Types of Contracts.
7. Selection of an Appropriate Approach in Project.
8. Software Effort Estimation.
9. Activity Planning: Project and Activities, Sequencings and Scheduling Activities.
10. Risk Management
11. Risk Control, RMMM, Configuration Management & Maintenance, Environment for Configuration Control, Configuration Control vs. Version Control
12. Resource Allocation.
13. Monitoring & Control.
14. Review and Evaluation.
15. Challenges of Outsourcing in Project Management, Presentations

Recommended Texts

1. Hughes, B., & Cotterell, M. (2009). *Software project management*. New Delhi: Tata McGraw-Hill Education.
2. Rose, K. H. (2013). *A guide to the project management body of knowledge*. Newtown Square*:* Project management Institute, Inc.

Suggested Readings

1. Stellman, A., & Greene, J. (2005). *Applied software project management*. California: "O'Reilly Media, Inc.".

SECC-6104 **Software Re-Engineering** **3(3+0)**

At the end of the course the students will be able to understand basic of software maintenance terminologies. Understand concept of maintenance effort using COCOMO model, analyzing the software scale drivers and software cost drivers. Explain legacy systems, legacy system components and the concept of software reuse in supporting software maintainability, List the concept of architectural evolution concept. Understand Reusability techniques and software metrics. Learning objectives of the course include: understanding the factors that make change of existing systems both technically challenging and risky, and the processes required to control change. Having a knowledge and understanding of the specific problems inherent in the reengineering and evolution of legacy software systems, and be able to apply some of the techniques that can be of use in comprehending and changing them. Having a knowledge and understanding of the specific challenges inherent in the reengineering and evolution of data-intensive systems, and be able to apply some of the techniques that can be of use in comprehending and changing them.

Contents

1. Software Maintenance.
2. Software Maintenance Process Models.
3. Analysis, System Release Planning, Change Implementation, Regression Testing and System Testing, Acceptance Testing, Quality Assurance, System Release, Version and Release management issues
4. The software maintenance process Models: Quick-and-Fix Model, Bohem's Model
5. Software Maintenance Process Models, SW Maintenance difficulties
6. Legacy Systems.
7. Legacy System Design, Legacy replacement strategies., Legacy System Assessment
8. Software Cost Modeling.
9. Software Cost Computing.
10. System Evolution; Program evolution dynamics, Architectural evolution (n-tire), Architectural evolution (VMC, SC, Web services).
11. Software Re- engineering in Maintenance.
12. Reverse Engineering Process, Program structure improvement, Program modularization.
13. Data re-engineering in Maintenance; Data re-engineering Process, Data Migration, Data Restructuring
14. Software Reuse and Reuse Landscape; Software Reusability Definition, Problems. Benefits, Approaches to Reuse, Software Reuse and Maintainability Issues Design Patterns, Frameworks, Program Generators, COTS, Reuse, Aspect-Oriented Development, Product Lines, Web-Services.
15. Software Metrics; Software Quality Measures, Types of measures.

Recommended Texts

1. Tripathy, P., & Naik, K. (2014). *Software evolution and maintenance: a practitioner's approach*. New York: John Wiley & Sons.
2. Sommerville, I. (2011). Software engineering (9^{th} ed.). New York: Addison Wesley.

Suggested Readings

1. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. New York: Palgrave

macmillan.
2. Penny, G. (2003). *Software maintenance: concepts and practice*. Singapore: World Scientific.

ENGL-5101 English composition & comprehension 3(3+0)

At the end of the course students can articulate clearly, take and pass on messages, deal with customers effectively, read, understand and follow a wide range of documents and write fluently and accurately, using accepted business conventions of format, spelling, grammar and punctuation, This course is developed to strengthen students these skills which enable them to deal with the practical problems and challenges of life – at home, in education and at work. This course focuses on critical reading and strategies for varying writing style, tone, and form for multiple purposes and audiences. Students learn to compose texts, including academic essays that clearly assert a claim and support the claim with compelling evidence. Other objectives of the course include: Compose a clear, coherent, unified essay organized around a single central idea and using a variety of techniques for support. Demonstrate understanding of and effective use of paragraph structure, including topic sentences, supporting examples and transition sentences.

Contents

1. Periods, Question Marks, Exclamation Marks, Semicolons
2. Colons, Commas, Apostrophes, Quotation Marks.
3. Writing Mechanics: Capitals, Abbreviations; Vocabulary: Frequently Confused Words, Frequently Misused Words, Vocabulary:
4. Phrases, Synonyms, Antonyms, Idioms, General Vocabulary,
5. Use of Articles and One, A Little/ A Few, This, That, Care, Like, Love, Hate, Prefer, Wish, All, Each, Every, Both, Neither, Either, Some, Any, No, None;
6. Interrogatives: Wh-? Words and How? Kinds of Nouns; Kinds of Adjectives;
7. Adverbs: Kinds, Form, Position and Use, Prepositions; Possessive, Personal and Reflexive Pronouns; Relative Pronouns and Clauses Classes of Verbs:
8. Ordinary Verbs, Auxiliary Verbs (Be, Have, Do); May and Can for Permission and Possibility; Can and Be Able for Ability; Ought, Should, Must, Have To, Need for Obligation; Must, Have, Will and Should for Deduction and Assumption;
9. The Auxiliaries Dare and Used.The Present Tenses the Past and Perfect Tenses,
10. The Infinitive Commands, Requests, Invitations, Advice, Suggestions,
11. The Subjunctive, The Passive Voice; Indirect Speech Conjunctions,
12. Purpose, Noun Clauses; Clauses of Reason, Result, Concession,
13. Comparison, Time Numerals, Dates, and Weights and Measures;
14. Spelling Rules,
15. Phrasal Verbs, List of Irregular Verbs

Recommended Texts

1. Thomson, A. J., & Martinet, A. V. (1980). *A practical english grammar*. Oxford: Oxford University Press.

Suggested Readings

1. Swan, M. (1984). *Basic english usage*. San Diego: Oxford University Press.
2. Nicanor L. Guintomary Ann R. Sibal Brian D. Villaverde. *(2014). Functional english in a global*

society: vocabulary building and communicative grammar. Luzon: Dept. of Lan-guages, Literature and Humanities College of Arts and Sciences Southern Luzon State University.

ENGL-5102 **Communication Skills** **3(3+0)**

At the end of the course the students will be able to sensitize students to their communicative behavior, to enable them to reflect and improve on their communicative behavior/performance to build capacities for self-criticism and facilitate growth to lead students to effective performances in communication. In this course students will learn about paragraph and essay writing to represent their ideas in clear thoughts. This course will demonstrate improved interpersonal skills by identifying and developing a repertoire of strategies for improved communication effectiveness and demonstrate the strategies in oral and written contexts. Students will be able to understand the research methods associated with the study of human communication, and apply at least one of those approaches to the analysis and evaluation of human communication. Students will develop knowledge, skills, and judgment around human communication that facilitate their ability to work collaboratively with others. Such skills could include communication competencies such as managing conflict, understanding small group processes, active listening, appropriate self-disclosure, etc.

Contents

1. Communication Skills
2. Internal Representation
3. Elements of Communication
4. Listening: Real Vs. Introduction
5. Expressing
6. Clarifying Language
7. Making Contact
8. Prejudgment: Prejudgment Traps, Stereotypes
9. Influencing Others
10. Public Speaking
11. Preparing A Formal Oral Presentation
12. Delivering Presentation
13. Interviewing
14. Effective Written Communication
15. Building Rapport

Pre-requisite: Nil

Recommended Texts

1. Training, M. T. D. (2012). *Effective communication skills*. London: Bookboon.

Suggested Readings

1. McKay, M., Davis, M., & Fanning, P. (2009). *Messages: The communication skills book*. New Harbinger Publications.
2. Dr. M. A. Pa-sha &Dr. S. Pasha. (2011). *Secrets of successful presenters: a guide for successful presenters.* New York: Lambert Academic Publishing.

ENGL-6101 **Technical and Report Writing** **3(3+0)**

Technical writing is a formal, structured and sophisticated writing to fulfill the requirements for a particular field of study. The course aims at providing understanding of writer's goal of writing (i.e. clear, organized and effective content) and to use that understanding and awareness for academic reading and writing. The objectives of the course are to make the students acquire and master the technical writing skills. The course would enable the students to develop argumentative writing techniques. The students would be able to the content logically to add specific details on the topics such as facts, examples and statistical or numerical values. The course will also provide insight to convey the knowledge and ideas in objective and persuasive manner. Furthermore, the course will also enhance the students' understanding of ethical considerations in writing academic assignments and topics including citation, plagiarism, formatting and referencing the sources as well as the technical aspects involved in referencing.

Contents

1. Writing for Readers: Academic, Public, and Work Communities; Analyzing Electronic Communities; Myths and Realities about Writing.
2. Discovering and Planning.
3. Purpose, Thesis, and Audience: Identifying Your Focus and Purpose; Creating a The-sis; Understanding Your Readers.
4. Drafting.
5. Revising, Editing, and Proofreading: Making Major Revisions; Making Minor Revisions; Revising Collaboratively; Revising in Digital Environments; Editing; Editing Collaboratively; Proofreading.
6. Paragraphs: Unfocused Paragraphs; Revising for Focus; Incoherent Paragraphs; Re-vising for Coherence; Poorly Developed Paragraphs; Revising for Development; Using Special-Purpose Paragraphs.
7. Clear and Emphatic Sentences: Unclear Sentences; Revising for Clear Sentences; Re-vising for Variety and Emphasis, Reasoning Critically.
8. Reading Critically, Arguing Persuasively, Designing Documents, Writing in Online Communities.
9. Speaking Effectively.
10. Academic Writing: Social and Natural Sciences
11. Abstract, Informative Report, Lab Report, Research Report,
12. Public Writing, Researching and Writing.

Pre-requisite: Nil

Recommended Texts

1. Anson, C. M., Schwegler, R. A., & Muth, M. F. (2005). *The longman writer's companion*. New York: Longman Publishing Group.
2. Pickett and Laster, *(1997)*. *Technical english: writing, reading, and speaking*. New York: Addison-Wesley Educational Publication.

Suggested Readings

1. Alred, Gerald, Charles T. Brusaw and Walter E. (2001). ***The Technical Writer 's***

Companion. Bedford/St: Martin's

URCP-5106 Pakistan Studies 2(2+0)

The course is designed to acquaint the students of BS Programs with the rationale of the creation of Pakistan. The students would be apprised of the emergence, growth and development of Muslim nationalism in South Asia and the struggle for freedom, which eventually led to the establishment of Pakistan. While highlighting the main objectives of national life, the course explains further the socio-economic, political and cultural aspects of Pakistan's endeavors to develop and progress in the contemporary world. For this purpose, the foreign policy objectives and Pakistan's foreign relations with neighboring and other countries are also included. This curriculum has been developed to help students analyses the socio-political problems of Pakistan while highlighting various phases of its history before and after the partition and to develop a vision in them to become knowledgeable citizens of their homeland. Have an in-depth understanding about the working and impacts of a few major civil engineering projects in Pakistan.

Contents

1. Contextualizing Pakistan Studies
2. Geography of Pakistan: Geo-Strategic Importance of Pakistan
3. Freedom Movement (1857-1947)
4. Pakistan Movement (1940-47)
5. Muslim Nationalism in South Asia
6. Two Nations Theory
7. Ideology of Pakistan
8. Initial Problems of Pakistan
9. Political and Constitutional Developments in Pakistan
10. Economy of Pakistan: Problems and Prospects
11. Society and Culture of Pakistan
12. Foreign Policy Objectives of Pakistan and Diplomatic Relations
13. Current and Contemporary Issues of Pakistan
14. Human Rights: Issues of Human Rights in Pakistan

Recommended Texts

1. Kazimi, M. R. (2007). *Pakistan studies*. Karachi: Oxford University Press.
2. Sheikh, Javed Ahmad (2004). *Pakistan's political economic and diplomatic dynamics*. Lahore: Kitabistan Paper Products.

Suggested Readings

1. Hayat, Sikandar (2016). *Aspects of pakistan movement*. Islamabad: National Institute of Historical and Cultural Research.
2. Kazimi, M. R (2009). *A concise history of pakistan*. Karachi: Oxford University Press.
3. Talbot, Ian (1998). *Pakistan: a modern history*. London: Hurst and Company.

URCI-5105 **Islamic Studies** **2(2+0)**

Islamic Studies engages in the study of Islam as a textual tradition inscribed in the fundamental sources of Islam; Qur'an and Hadith, history and particular cultural contexts. The area seeks to provide an introduction to and a specialization in Islam through a large variety of expressions (literary, poetic, social, and political) and through a variety of methods (literary criticism, hermeneutics, history, sociology, and anthropology). It offers opportunities to get fully introductory foundational bases of Islam in fields that include Qur'anic studies, Hadith and Seerah of Prophet Muhammad (PBUH), Islamic philosophy, and Islamic law, culture and theology through the textual study of Qur'an and Sunnah. Islamic Studies is the academic study of Islam and Islamic culture. It majorly comprises of the importance of life and that after death. It is one of the best systems of education, which makes an ethical groomed person with the qualities which he/she should have as a human being. The basic sources of the Islamic Studies are the Holy Qur'an and Sunnah or Hadith of the Holy Prophet Muhammadﷺ. The learning of the Qur'an and Sunnah guides the Muslims to live peacefully.

Contents

1. Study of the Qur'an (Introduction to the Qur'an, Selected verses from *Surah Al-Baqarah, Al-Furqan, Al-Ahzab, Al-Mu'minoon, Al-An'am, Al-Hujurat, Al-Saff*)
2. Study of the Hadith (Introduction to Hadith literature, Selected Ahadith (Text and Translation)
3. Introduction to Qur'anic Studies
4. Basic Concepts of Qur'an
5. History of Quran
6. Basic Concepts of Hadith
7. History of Hadith
8. Kinds of Hadith
9. Uloom –ul-Hadith
10. Sunnah & Hadith
11. Seerat ul-Nabi (PBUH), necessity and importance of Seerat, role of Seerah in the development of personality, Pact of Madinah, Khutbah Hajjat al-Wada' and ethical teachings of Prophet (PBUH).
12. Legal Position of Sunnah
13. Islamic Culture & Civilization
14. Characteristics of Islamic Culture & Civilization
15. Historical Development of Islamic Culture & Civilization
16. Comparative Religions and Contemporary Issues
17. Impact of Islamic civilization

Recommend Texts

1. Hassan, A. (1990). Principles of islamic jurisprudence. New Dehli: Adam Publishers.
2. Zia-ul-Haq, M. (2001). Introduction to al-sharia al-islamia. Lahore: Aziz Publication.

Suggested Readings

1. Hameedullah, M. (1957). *Introduction to islam*. Lahore: Sh M Ashraf Publisher.

2. Hameedullah, M. (1980). *Emergence of islam*. New Dehli: Adam Publishers.
3. Hameedullah, M. (1942). *Muslim conduct of state*. Lahore: Sh M Ashraf Publisher.

ICTC-5201 Introduction to Information & Communication Technologies 3(2+1)

The course introduces students to information and communication technologies and their current applications in their respective areas. Objectives include basic understanding of computer software, hardware, and associated technologies. They can make use of technology to get maximum benefit related to their study domain. Students can learn how the Information and Communications systems can improve their work ability and productivity. How Internet technologies, E-Commerce applications and Mobile Computing can influence the businesses and workplace. At the end of semester students will get basic understanding of Computer Systems, Storage Devices, Operating systems, E-commerce, Data Networks, Databases, and associated technologies. They will also learn Microsoft Office tools that includes Word, Power Point, Excel. They will also learn Open office being used on other operating systems and platforms. Specific software's related to specialization areas are also part of course. Course will also cover Computer Ethics and related Social media norms and cyber laws.

Contents

1. Introduction to Computers.
2. The Internet and World Wide Web.
3. Application Software.
4. The System Unit.
5. Input Device.
6. Output Devices,
7. Storage.
8. System Software.
9. Communications.
10. Databases.
11. Computer Security and Safety, Ethics, and Privacy.

Pre-requisite: Nil

Recommended Texts

1. Vermaat, M. E. (2018). *Discovering computers: digital technology, data and devices.* Boston: Course Technology Press.

Suggested Readings

1. Timothy J. O'Leary & Linda I. (2017). *Computing essentials*. San Francisco: McGraw Hill Higher Education.
2. Schneider, G. M., & Gersting, J. (2018). *Invitation to computer science*. Boston: Cengage Learning.

SESC-6101 **Professional Practices** **3(3+0)**

This course is a graduate-level course of Professional Practices. The course aims to elaborate foundation of Professional Practices as a subject. It focuses on both mainstream and critical approaches to visualize and examining how these topics conceptualize Professional Practices as a field of study. The course explicitly relates Professional Practices as cognate. Students will gain confidence in their ability to communicate by practicing and receiving feedback on business communication skills. Through this critical enquiry process, it is expected that students will further develop their understandings of their own practices and change and develop aspects of these practices. It has a particular emphasis on building professional knowledge of working in a business environment and developing work ready skills in the areas of written and oral communication, intercultural communication, client service, problem solving and self-management. Students will develop strategies to communicate these abilities through career communication, including career documents and interviews. Opportunities for improving academic and workplace language proficiency are embedded in the course.

Contents

1. The Engineering Profession.
2. The Structure of Organizations.
3. Finance and Accounting.
4. Anatomy of a Software House.
5. Computer Contracts.
6. Intellectual Property Rights.
7. The Framework of Employee Relations Law and Changing Management Practices.
8. Human Resource Management and Software Engineering.
9. Health and Safety at Work.
10. Software Liability: Liability and Practice.
11. Computer Misuse and the Criminal Law.
12. Graphical Methods for Comparing Means.
13. Regulation and Control of Personal Information: Data Protection, Defamation and Related Issues.
14. The British Computer Society Code of Conduct.

Pre-requisite: Nil

Recommended Texts

1. Bott, F., Coleman, A., & Rowland, D. (2000). *Professional issues in software engineering*. New York: CRC Press.

Suggested Readings

1. Sara Baase. *A gift of fire: social, legal, and ethical issues for computing and the internet* (3rd ed). Harlow: Prentice Hall
2. Rost, J., & Glass, R. L. (2011). *The dark side of software engineering: evil on computing projects*. New Jersey: John Wiley & Sons.
3. Jones, C. (2009). *Software engineering best practices*. New York: McGraw-Hill, Inc.

MATH-5101 **Calculus and Analytical Geometry** **3(3+0)**

This course provides the foundation and basic ground for calculus and analytical geometry background. Understand the foundation and basic ground for calculus and analytical geometry background. To learn fundamentals of mathematics, calculus and analytical geometry. To enable students to apply the ideas to solve problems of practical nature. Have knowledge related to the fundamentals of calculus and analytical geometry. Purpose of this course is to build the student's knowledge of differential/integral calculus of multi-variable functions based on their past experience of differential/integral calculus and analytic geometry of functions of one independent variable. Students will Understand the differentiation integration and their applications. Apply the acquired knowledge to solve problems of practical nature. After the successful completion of course, the students will be able to familiar to real value functions of one and several variables, will learn to analyze and solve problems relating analytical geometry, vector analysis & vector calculus and initial value problems.

Contents

1. Real Numbers and the Real Line, Coordinates, Lines, and Increments, Functions, Shifting Graphs, Trigonometric Functions.
2. Limits and Continuity.
3. Derivatives, The Chain Rule.
4. Applications of Derivatives: Extreme V, alues of Functions, The Mean Value Theorem, The First Derivative Test for Local Extreme Values.
5. Integration, Substitution in Definite Integrals. Numerical Integration.
6. Applications of Integrals: Areas between Curves, Finding Volumes by Slicing, Volumes of Solids of Revolution—Disks and Washers. Cylindrical Shells. Lengths of Plan Curves, Areas of Surfaces of Revolution, Moments and Centers of Mass.
7. Transcendental Functions: Inverse Functions and Their Derivatives, Natural Logarithms, The Exponential Function, ax and logax, Growth and Decay, L'Hôpital's Rule, Relative Rates of Growth.
8. Conic Sections, Parameterized Curves, and Polar Coordinates: Conic Sections and Quadratic Equations. Classifying Conic Sections by Eccentricity. Quadratic Equations and Rotations.
9. Vectors and Analytic Geometry in Space, Vectors in the Plane Dot Products, Vector-Valued Function Cartesian (Rectangular) Coordinates and Vectors in Space. Dot Products. Cross Products. Lines and Planes in Space. Cylinders and Quadric Surfaces. Cylindrical and Spherical Coordinates.

Pre-requisite: Nil

Recommended Texts

1. George B. Thomas and Ross L. Finney. *calculus and analytic Geometry (*10th ed.). New York: Addison Wesley

Suggested Readings

1. Swokowski, Olinick and Pence. (1979). *Calculus and Analytical Geometry*. (6th ed.). Berlin: Brooks/Cole Publishers.
2. Anton, H., & Herr, A. (1988). *Calculus with analytic geometry*. New York: Wiley.

MATH-5102 **Linear Algebra** **3(3+0)**

This course provides the fundamentals solution for the system of linear equations, operations on system of equations, matrix properties, solutions and study of their properties. Students completing this course will be able to compute the inverse of an invertible matrix. Students completing this course will be able to find the null space of a matrix and represent it as the span of independent vectors. Students completing this course will be able to find the matrix representation of a linear transformation given bases of the relevant vector spaces. Computer software and graphing calculators will be used to enhance the learning and teaching of topics and techniques covered. This course will be creating the abilities in the students to compute the inverse of an invertible matrix. After completing this course students will be able to find the null space of a matrix and represent it as the span of independent vectors.

Contents

1. Introduction to Vectors: Lengths and Dot Products, Matrices.
2. Solving Linear Equations: Vectors and Linear Equations, the Idea of Elimination, Elimination Using Matrices, Rules for Matrix Operations, Inverse Matrices.
3. Elimination = Factorization; A = LU, Transposes and Permutations
4. Vector Spaces and Subspaces: Spaces of Vectors, The Null space of A: Solving Ax = 0, The Rank and the Row Reduced Form, the Complete Solution to Ax = B, Independence, Basis and Dimension, Dimensions of the Four Subspaces.
5. Orthogonally: Orthogonally of the Four Subspaces, Projections, Least Squares Approximations, Orthogonal Bases and Gram-Schmidt.
6. Determinants: The Properties of Determinants, Permutations and Cofactors, Cramer's Rule, Inverses, and Volumes.
7. Eigenvalues and Eigenvectors: Introduction to Eigenvalues, Diagonalizing a Matrix, Applications to Differential Equations, Symmetric Matrices, Positive Definite Matrices, Similar Matrices, Singular Value Decomposition (SVD).
8. Applications: Matrices in Engineering, Graphs and Networks, Markov Matrices, Population, and Economics; Linear Programming, Fourier series: Linear Algebra for Functions, Linear Algebra for Statistics and Probability, Computer Graphics.
9. Numerical Linear Algebra: Gaussian Elimination in Practice, Norms and Condition Numbers, Iterative Methods for Linear Algebra.
10. Complex Vectors and Matrices: Complex Numbers, Hermitian and Unitary Matrices, Matrix Factorizations.

Pre-requisite: Nil

Recommended Books

1. Strang, G., Strang, G., Strang, G., & Strang, G. (1993). *Introduction to linear algebra* (Vol. 3). Wellesley, MA: Wellesley-Cambridge Press.
2. Advanced Zill, D., Wright, W. S., & Cullen, M. R. (2011). *Advanced engineering mathematics*. New York: Jones & Bartlett Learning.

Suggested Books

1. Humphreys, John F. *A (1996). Course on group theory*. Oxford: Oxford University Press

MATH-5103 **Probability and Statistics** **3(3+0)**

At the end of the course the students will be able to understand the concepts of data analysis, presentation, counting techniques, probability and decision making. It is important because of its direct application in areas such as genetics, finance and telecommunications. It also forms the fundamental basis for many other areas in the mathematical sciences including statistics, modern optimization methods and risk modelling. Students who successfully complete this course should be able to demonstrate understanding of: basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables. How to derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions. They will know how to calculate probabilities, and derive the marginal and conditional distributions of bivariate random variables. Students may able to grasp other concepts like mathematical expectation, sampling etc.

Contents

1. Introduction to Statistics and Data Analysis
2. Discrete and Continuous Data, Statistical Modeling, Scientific Inspection, and Graphical, General Types of Statistical Studies
3. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability
4. Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions
5. Mathematical Expectation.
6. Discrete Probability Distributions.
7. Continuous Probability Distributions.
8. Fundamental Sampling Distributions and Data Descriptions.
9. One- and Two-Sample Estimation Problems.
10. Single Sample: Estimating a Proportion, Estimating the Difference between Two Proportions, Single Sample: Estimating the Variance, Estimating the Ratio of Two Variances.
11. One- and Two-Sample Tests of Hypotheses.
12. Single Sample: Tests Concerning a Single Mean, Two Samples: Tests on Two Means, Choice of Sample Size for Testing Means, Graphical Methods for Comparing Means, One Sample: Test on a Single Proportion, Two Samples: Tests on Two Proportions.

Pre-requisite: Nil

Recommended Texts

1. Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (1993). *Probability and statistics for engineers and scientists* (Vol. 5). New York: Macmillan.

Suggested Readings

1. Anthony J. Hayter, *Probability and statistics for engineers and scientists*. New York: Duxbury Press.
2. Haigh, J. (2012). *Probability: A very short introduction* (Vol. 310). Oxford: Oxford University Press.

PHYS-5101 **Applied Physics** **3(3+0)**

This course develops the understanding of the fundamental concepts/laws in physics by explaining and discussing the physics as well as their relevance to everyday events and circumstances in a broad interdisciplinary context. It demonstrates teamwork skills/ ability to collaborate by working in groups on a laboratory experiment. At the end of the course the students will be able to understand the basic laws of physics, circuiting and basics of transistors. The primary objective of the course is to endow the knowledge of a wide variety of electric and magnetic phenomena along with their scientific applications, specifically, in the field of computer science. This course equips the students with the applied concepts of the Physics. Course brushes the basic knowledge of students by starting from the basic concepts and then progresses gradually toward the advance concepts. At the end of the course, students would have developed good understanding of Physics fundamentals.

Contents

1. Zero Reference Level, Chassis Ground, Ohm ‘s Law, Formula Variations of Ohm ‘s Law, Graphical Representation of Ohm ‘s Law, Linear Resistor, Non-Linear Resistor, Cells in Series and Parallel.
2. Resistive Circuits.
3. Resistors, Inductors Capacitors
4. Energy Sources.
5. Magnetism and electromagnetism.
6. Solid State. Atomic structure, Electron distribution of different atoms, Energy bands in solids, Bonds in solids, Conduction in solids, Conductors, Semiconductors and types of semiconductors, Insulators, Majority and Minority charge carriers, Mobile charge carriers and immobile ions, Drift current in good conductors.
7. P-N Junction. Formation of depletion layer, Junction or barrier voltage, Forward biased P-N Junction, Forward V/I Characteristics, Reverse biased P-N Junction, Re-verse Saturation Current, Reverse V/I Characteristics, Junction breakdown, Junction Capacitance.
8. Optoelectronics Devices. Spectral response of human eye, Light Emitting Diode (LED), Photoemissive Devices, Photomultiplier Tube, Photovoltaic Devices, Bulk type Photoconductive cells, Photodiodes, P-N junction Photodiode, PIN Photodiode, and Avalanche Photodiode.
9. DC Power Supplies. Unregulated and Regulated Power Supply, Steady and Pulsating DC Voltages, Rectifiers (17.5-17.8), Filters (17.9-17.2), Voltage Multipliers (17.24-17.30), Silicon Controlled Rectifier SCR (17.33-17.37)
10. The Basic Transistor. Transistor Biasing, Transistor Circuit Configuration.
11. Modulation and Demodulation, Integrated Circuitsr, Fiber Optics.

Pre-requisite: Nil

Recommended Texts

1. Streetman, B. G., & Banerjee, S. (1995). *Solid state electronic devices* (Vol. 4). New York: Prentice hall.

Suggested Readings

1. Malvino, A. P., & Tipler, P. A. (2001). *Electronic principles*. Glencoe: MacGraw-Hill.

SESC-6101 **Business Process Engineering** **3(3+0)**

At the end of the course the students will be able to discover the various technologies that support Business Process Engineering. They will be able to learn and analyze the performance of existing processes and identify process improvement. Propose business solutions in written and verbal forms for process innovation and redesign projects. Create a BPM implementation strategy and implementation plan for an organization. It illustrates how to take giant strides to attain market dominance in a dynamic business world. The course shows how automation of re-engineered processes can increase competitive advantage for a firm. Using several case studies, the course also shows how different companies have streamlined their processes, reduced their cost of operations, have created cross functional process excellence to increase value proposition to all stakeholders. At the end of the course students may able to understand the evolution of BPR, BPR methodologies, understand the Research Methodology, will have the ability to analyze a practical application of the BPR Framework.

Contents

1. Introduction to Business Process Engineering, Motivation and Definitions, Business Process Lifecycle.
2. Classification of Business Processes, Goals, Structure, and Organization
3. Business Reengineering for Information Technology
4. Evolution of Enterprise Systems Architectures.
5. Market-and Product-Oriented Design of Business Processes
6. BPR Methodologies: Methods and Tools
7. Knowledge-Based Reengineering of Business Processes
8. Defense Enterprise Planning and Management
9. Supporting Business Process Reengineering in Industry Week1
10. Supporting Business Process Reengineering in Industry Week2
11. Dynamic Simulation in Business Process Redesign Week1
12. Dynamic Simulation in Business Process Redesign Week2
13. Various Approaches to Engineering
14. Networking as an Enabler of Business Process Reengineering

Pre-requisite: Nil

Recommended Texts

1. Elzinga, D. J., Gulledge, T. R., & Lee, C. Y. (2012). *Business process engineering: Advancing the state of the art*. New York: Springer Science & Business Media.

Suggested Readings

1. Scheer, A. W. (2012). *Business process engineering: reference models for industrial enterprises*. New York Springer Science & Business Media.
2. Unhelkar, B. (Ed.). (2009). *Collaborative business process engineering and global organizations: frameworks for service integration: frameworks for service integration*. New York: IGI Global.

SESC-6102 Formal Methods in Software Engineering 3(3+0)

This course provides a hands-on introduction to formal methods for software engineering. The purpose of formal methods is to enable the construction of highly reliable software. Formal methods are concerned with specifications that are precise for being stated in languages endowed with a formal syntax, semantics, and theory. The course builds on skills in first-order logic and temporal logic, and shows how these formalisms can be applied, and extended, for the verification of software. At the end of the course the students will be able to describe the costs and benefits of formal methods. Construct formal models of sequential software systems. Students may also be able to implement sequential software systems based on formal models. Verify attributes of formal models, and demonstrate formal correctness of simple procedure. Students may able to learn concepts like the Z (Zed) specification Language, Z Schemas and Schema Calculus, Connectives, Logic and Natural Language, formal reasoning etc.

Contents

1. Formal methods: What is FM? What FM are not, When, How, and Why use FM? Popular Fallacies and Alternatives. Formal Methods and Project Management: Gathering Requirement, From Information Requirement to Formal Specifications.
2. Introducing Z.
3. Elements of Z.
4. Connectives, Logic and Natural Language, Quantifiers, Z and Boolean Types, Predicates and undefined Expressions. Synthesis: Set Comprehensives, Lambda Expressions, Formal Specifications, Conveniences and shortcuts, Modeling Systems and Change.
5. Schemas and schema calculus: Conjunctions and Disjunctions, Other Schema Operators.
6. Formal Reasoning: Calculation and proof, Laws, Checking Specifications, Preconditions, Formal Reasoning and Intuition, Machine-Checked proof.
7. Studies in Z: Document Control System, Text Processing, Eight Queens.
8. Computer Graphics and Computational Geometry. Rule-Based Programming
9. Graphical User Interface.
10. Safety-Critical Protection System.
11. Modeling Large Systems.
12. Object-Oriented Programming Model and Z, Inherits and Schema Inclusion, OO Z Dialects. Concurrency and Real-time.
13. Refinement, Program Derivation and Formal Verification.
14. Converting Z Specification into Code

Recommended Texts

1. Jacky, J. (1997). *The way of Z: practical programming with formal methods*. New York: Cambridge University Press.

Suggested Readings

1. Huth, M., & Ryan, M. (2004). *Logic in computer science: modelling and reasoning about systems*. New York: Cambridge university press.
2. Gabbar, H. A. (Ed.). (2006). *Modern formal methods and applications*. Netherlands: Springer Science & Business Media.

SESC-6103 **Cyber Security** **3(3+0)**

At the end of the course the students will be able to describe the core information assurance (IA) principles. Identify the key components of cyber security architecture. Distinguish system and application security threats and vulnerabilities. Define types of incidents including categories, responses and timelines for response. In the course students will do assignments focused on introductory concepts in cybersecurity. These concepts include cybersecurity theory and basic techniques for optimizing security on personal computers and small networks. Learning outcomes of the course include: to describe the role of computers and networks in a security context; identify computer system threats and evaluate their impact; discuss the effectiveness of various cryptographic techniques and their impact on security; develop basic organizational security policies; and demonstrate how defense in depth can be used to implement security. At the end of the course students may able to learn about cyberspace laws, different security tools available and how to implement basic security strategies to a network or computer.

Contents

1. Vulnerabilities in information system, measuring vulnerabilities.
2. Threat classification, cyber security starts at home and international awareness.
3. Vulnerabilities in the organization, access authorization and authentication, security services in wireless networks and cloud security.
4. Risk in information system infrastructure, hardware, software, and cyberspace.
5. Assets identification, resource access control and securing the assets communication.
6. Secure information system, information security management.
7. Cyber security and the CIO, data backup and archiving, cyber trainings and cyber policy.
8. Building a secure organization, system access control and computer network management securely.
9. Personal, physical and environmental security and business continuity planning.
10. Cyberspace Intrusions, ID/PS configuration, ID/PS management and ID/PS classification.
11. ID/PS implementation and operation in organization.
12. Cyberspace defense, file protection application, PC performance applications.
13. Protection tools, security analyzer, password analyzer, firewalls and email protection.
14. Cyberspace and law, international law and cyber related laws.
15. Cybercrime, trends in cyber abuse, combating cybercrime.
16. Cyber warfare and homeland security and distributed defense.

Pre-requisite: Nil

Recommended Texts

1. Kostopoulos, G. (2017). *Cyberspace and cybersecurity*. New York: CRC Press.

Suggested Readings

1. Ciampa, M. (2012). *Security+ guide to network security fundamentals*. New York: Cengage Learning.
2. Nam Nguyen (2016). *Essential cyber security handbook kindle edition.* New York. CRC Press.

BUSB-5101 **Principles of Management** **3(3+0)**

Formally defined, the principles of management are the activities that “plan, organize, and control the operations of the basic elements of [people], materials, machines, methods, money and markets, providing direction and coordination, and giving leadership to human efforts, so as to achieve the sought objectives of the enterprise. This course presents the principles, techniques, and concepts needed for managerial analysis and decision-making. It highlights the effective management of planning, organizing, influencing, and controlling related to the internal and external environment and issues of ethics and social responsibility. Students will examine the fundamental roles and process of planning, leading, organizing, and controlling that comprise the manager’s role. The course will cover topics fundamentals and principles of management, administrative policy, objectives, and procedures and problems of organizational control and leadership. The course will enable students to develop short- and long-term plans to effectively accomplish organization goals. Students will develop skills related to the manager’s functions as required in today’s competitive environment.

Contents

1. Introduction to Managers and Management.
2. Organizational Culture and Environment.
3. Decision Making the Essence of Manager’s Job: The Decision-Making Process
4. Planning.
5. Organization Structure and Design
6. Motivation
7. Leadership
8. Communication
9. Controlling
10. The Personnel Function
11. Job Design and Analysis
12. Human Resource Planning
13. Recruitment and Selections/Testing and Interview
14. Miscellaneous: Union and Management, Compensation Administration, Health and Safety

Pre-requisite: Nil

Recommended Texts

1. Robbins, S.P. & Coulter, Mary. (2008). *Management,* (10th ed.). Mary: Prentice Hall.
2. Robbins, S.P. & DeCenzo, David A. (2010). *Fundamentals of management*, (7th ed.). New York: Prentice Hall.

Suggested Readings

1. DeCenzo, D. A., Robbins, S. P., & Verhulst, S. L. (2016). *Fundamentals of human resource management*. New York: John Wiley & Sons.
2. Charles W. L. Hill and McShane S. (2006). *Principles of management* (1st ed.). New York: McGraw-Hill

3. Carpenter M. (2009). *Principles of management*. New York: Flat World Knowledge.
4. Richard L. Daft. (2011). *management* (10^{th} ed.). California: South-Western College Pub

BUSB-5107 Human Resource Management 3 (3+0)

The course is designed to impart knowledge about the meaning, scope, objectives, and significance of human resources development and management in 21st-century organizations. The principal objective of this course is to prepare students to gain knowledge in the field of human resource management, enabling them to understand the human resource functions and practices in organizations for improved performance and to help create a transparent organizational culture. In this course, effective human resource management policies, practices, and procedures are examined. Issues and strategies required to select and develop manpower resources, workplace health and safety programs/policies are discussed. Students will be able to identify the human resources needs of an organization, conduct a job analysis, and produce a job description. The course also provides an overall understanding of organizational behavior concepts to assist students in recognizing organizational structure, culture, and development concepts so that they are better equipped to perform in the organization, make informed decisions and effectively manage supervisors and subordinates for enhanced performance.

Contents

1. Managing human resources
2. Understanding the external and organizational environments
3. Ensuring fair treatment and legal compliance
4. HR planning for alignment and change
5. Using job analysis and competency modeling
6. Recruiting and retaining qualified employees
7. Selecting employees to fit the job and the organization
8. Training and developing a competitive workforce
9. Conducting performance management
10. Developing an approach to total compensation
11. Using performance-based pay to achieve strategic objectives
12. Providing benefits and services for employee's well-being
13. Risk management, employee relations, and risk management
14. Health, safety, and employee well-being
15. Understanding un-ionization and collective bargaining

Recommended Texts

1. Jackson, S., Schuler, R. & Werner, S. (2012). *Managing human resources*. Mason OH: South Western/Cengage Learning.

Suggested Readings

1. Mathis, R. & Jackson, J. (2011). *Human resource management*. Mason, OH: Thomson/South-western.
2. Dessler, G., Cole, N. & Munro, C. (2011). *Management of human resources*. Toronto: Pearson Canada.

BUSB-5103 **Organizational Behavior** **3 (3+0)**

This course is concerned with the study of the behavior of individuals and groups as part of the social and technical system in the workplace. The course discusses the individual and group behavior, communication, conflict, and various management styles, motivational techniques, and coordination in the work environment. Conceptual frameworks, case discussions, and skill-oriented activities are applied to course topics which include: values, social perceptions, motivation, learning and development, group dynamics, leadership, power, influence, politics, change, diversity, communication, conflict management, organizational design, and innovation. In this course, the students will learn about the different motivational theories and evaluate motivational strategies used in a variety of organizational settings. This course also provides a theoretical foundation that can be used to analyze individual and group behavior, assess the organizational structure, and evaluate various managerial practices and leadership styles in the organization. Furthermore, students will acquire skills and analytic concepts to improve organizational relationships, effectively respond to the emerging needs of the organization, and to enhance the organization's human resources.

Contents

1. Organizational behavior: the quest for people-centered organizations and ethical conduct
2. Organizational culture, socialization, and mentoring
3. Key individual differences and the road to success
4. Values, attitudes, job satisfaction, and counterproductive work behaviors
5. Social perceptions and attributions
6. Foundations of motivation
7. Improving job performance with goals, feedback, rewards, and positive reinforcement
8. Group dynamics
9. Developing and leading effective teams
10. Individual and group decision making
11. Managing conflict and negotiating
12. Communicating in the digital age
13. Leadership, influence, empowerment, and politics
14. Organizational design, effectiveness, and innovation

Pre-requisite: Nil

Recommended Texts

1. Kreitner, R. & Kinicki, A. (2013). *Organizational behavior*. New York: McGraw-Hill/Irwin.

Suggested Readings

1. Robbins, S. & Judge, T. (2013). *Organizational behavior*. Boston: MA: Pearson.
2. Johnson, C. (2012). *Meeting the ethical challenges of leadership: casting light or shadow*. Thousand Oaks, CA: SAGE.

BUSB-5104 **Business Economics** **3 (3+0)**

Business economics covers a broad set of topics relevant to management decisions, but microeconomics provides the foundation of its conceptual framework. This course is mean for understanding the concepts and principles of the economic environment within the businesses. The course focuses on three major topics: production theory, market structure, and pricing strategy. The impact of governmental regulations, managerial policies, and global issues affecting business are discussed. It provides an introduction to economics, decision-making in business, and the effects of policy on the broader economic environment. The goal of this course is to develop the skills necessary to make optimal managerial decisions given different situations, environments, and information sets by analysis and application of suitable economics concepts and principles. Students will be able to understand and apply supply and demand analysis, understand the causes and consequences of different market structures, and develop an understanding of national macroeconomic policy and the global trading environment.

Contents

1. Introduction to economics and its history
2. Basic concepts of the economic environment
3. Needs, wants, demand and consumer
4. Market and its environment
5. Introduction to business and the economic environment
6. The working of competitive markets
7. Demand and the consumer
8. Supply decisions in a perfectly competitive market
9. Pricing and output decisions in imperfectly competitive markets
10. Multinational environment
11. Business growth and strategy
12. Multinational corporations and business strategy in a global economy
13. The Government, the firm, and the market
14. The economy and business activity
15. National macroeconomic policy
16. The global trading environment

Pre-requisite: Nil

Recommended Texts

1. Sloman, J., Garratt, D., Guest, J., & Jones, E. (2016). *Economics for business*. Harlow: Pearson Education.

Suggested Readings

1. Griffiths, A. & Wall, S. (2008). *Economics for business and management*. Harlow, England New York: FT Prentice Hall.
2. McAleese, D. (2004). *Economics for business: competition, macro-stability, and globalisation*. Harlow, England New York: FT Prentice Hall.

BUSB-6101 **Entrepreneurship** **3 (3+0)**

This course provides exposure to both the fundamentals of business innovation and the practical aspects of identifying, evaluating, and moving the business ideas forward. It gives students the necessary tools to think creatively, to plan out whether their idea is marketable to investors, and guides them through the launch of their own business. Entrepreneurship is an interdisciplinary course designed to teach students how to think and act entrepreneurially. It provides an understanding of the entrepreneurship process and exposes the students to the concept, practices, and tools of the entrepreneurial world. It enables students to learn the characteristics of the 'entrepreneurial mindset' by understanding the differences between managerial and entrepreneurial thinking and develops the analytical skills of students with regard to various stages of the entrepreneurship process. Students gain an understanding of securing capital, planning, how to start-up, operate, and grow a business while in school, thus turning their learning into earning.

Contents

1. Entrepreneurship and the entrepreneurial mind-set
2. Entrepreneurial intentions and corporate entrepreneurship
3. Entrepreneurial strategy: generating and exploiting new entries
4. Creativity and the business idea
5. Identifying and analyzing domestic and international opportunities
6. Intellectual property and other legal issues for the entrepreneur
7. The business plan: creating and starting the venture
8. The marketing plans
9. The organizational plan
10. The financial plan
11. Sources of capital
12. Informal risk capital, venture capital, and going public
13. Strategies for growth and managing the implication of growth
14. Succession planning and strategies for harvesting and ending the venture

Pre-requisite: Nil

Recommended Texts

3. Hisrich, R., Peters, M. & Shepherd, D. (2013). *Entrepreneurship*. New York: McGraw-Hill/Irwin.
4. Greene, C. (2012). *Entrepreneurship: ideas in action*. Mason, OH: South-Western Cengage Learning.

Suggested Readings

1. Bygrave, W. & Zacharakis, A. (2011). *Entrepreneurship*. Hoboken, NJ: Wiley.
2. Kuratko, D. (2009). *Entrepreneurship: theory, process, practice.* Mason, Ohio: South-Western Cengage Learning.
3. Barringer, B. R. (2015). *Entrepreneurship: successfully launching new ventures*. Noida: Pearson Education India.

URCC-5110 Citizenship Education and Community Engagement 3 (1+2)

In recent years, community engagement has become a central dimension of governance as well as policy development and service delivery. However, efforts to directly involve citizens in policy processes have been bedeviled by crude understandings of the issues involved, and by poor selection of techniques for engaging citizens. This course will provide a critical interrogation of the central conceptual issues as well as an examination of how to design a program of effective community engagement. This course begins by asking: Why involve citizens in planning and policymaking? This leads to an examination of the politics of planning, conceptualizations of "community", and the tension between local and professional knowledge in policy making. This course will also analyze different types of citizen engagement and examine how to design a program of public participation for policy making. Approaches to evaluating community engagement programs will also be a component of the course. Moreover, to secure the future of society, citizens must train younger generations in civic engagement and participation. Citizenship education is education that provides the background knowledge necessary to create an ongoing stream of new citizens participating and engaging with the creation of a civilized society.

Contents

1. Introduction to citizenship education and community engagement
2. Introduction to active citizenship: overview of the ideas, concepts, philosophy and skills
3. Identity, culture and social harmony: concepts and development of identity
4. Components of culture and social harmony, cultural and religious diversity
5. Multi-cultural society and inter-cultural dialogue: bridging the differences, promoting harmony
6. Significance of diversity and its impact, importance and domains of inter-cultural harmony
7. Active citizen: locally active, globally connected
8. Importance of active citizenship at national and global level
9. Understanding community, identification of resources (human, natural and others)
10. Constitutionalism and citizen's responsibilities
11. Introduction to human rights, human rights in constitution of Pakistan
12. Universalism vs relativism, Public duties and responsibilities
13. Social Issues in Pakistan: introduction to the concept of social problem, causes and solutions
14. Social Issues in Pakistan: poverty, equal and equitable access of resources, unemployment
15. Social Issues in Pakistan: agricultural problems, terrorism & militancy, governance issues
16. Social action and project: introduction and planning of social action project
17. Identification of problem, rthical considerations related to project
18. Assessment of existing resources

Pre-requisite: Nil

Recommended Texts

1. Kennedy, J. K. Brunold, A. (2016). *Regional context and citizenship education in asia and europe*. New York: Routledge Falmer.

Suggested Readings

1. British, Council. (2017). *Active citizen's social action projects guide.* Scotland: British Council
2. Macionis, J. J. Gerber, M. L. (2010). *Sociology*. New York: Pearson Education

SEEC-6101 Model-Driven Software Development 3 (3+0)

Model-Driven Software Development (MDSD) is currently a highly regarded development paradigm among developers and researchers. MDSD is about using domain-specific languages to create models that express application structure or behaviour in an efficient and domain-specific way. These models are subsequently transformed into executable code by a sequence of model transformations. Modelling plays an important role in the development of large and complex software systems. Models are not only used to specify requirements and design for such systems but also to develop and communicate the understanding of a broad range of subject matters surrounding their development, operation, and maintenance. This course offers A comprehensive overview of MDSD and how it relates to industry standards such as MDA and Software Factories. Topics of the software development process, engineering issues such as versioning, testing, and product line engineering are also discussed. At the end of the course, students will be able to understand the different types of architecture and their role in the software systems. They will learn to develop models and check them for specific properties using modeling tools.

Content

1. Basic Ideas and terminology and overview of MDA concepts
2. Model-driven architecture, architecture-centric architecture, generative programming
3. Software factories, model-integrated computing, language-oriented programming
4. Classification: MDSD, CASE, 4GL, roundtrip engineering, domain-driven design, etc.
5. Meta-modeling
6. MDSD-capable target architectures
7. Building domain architecture
8. Code generation techniques: categorization, generation techniques
9. Model transformations with QVT
10. MDSD tools: roles, architecture, selection criteria, and pointers
11. The MDA standard: goals, core concepts, UML 2.0, MOF-meta object facility
12. MDSD process building block and best practices
13. Testing: types, test in model-driven application development, testing the domain architecture
14. Versioning: projects, dependencies, structure of application projects, version management

Pre-requisite: Nil

Recommended Texts

1. Stahl, T. & Völter, M. (2006). *Model-driven software development: technology, engineering, management*. New Jersey: J. Wiley & Sons.

Suggested Readings

1. Brambilla, M., Cabot, J. & Wimmer, M. (2012). *Model-driven software engineering in practice*. San Rafael: Morgan & Claypool.

SEEC-6102 **Software Engineering Economics** **3 (3+0)**

The course aims to elaborate the foundation of software engineering economics as a subject. Software engineering economics deals with decision making related to software engineering in a business context. The success of any software engineering project is partly dependent on effective business management. Software engineering economics provides a way to examine the attributes of software and software processes in a systematic way that relates them to economic measures. These attributes can be weighted and analyzed when making decisions within the scope of a software engineering project and its organization. The essence of software engineering economics is aligning software technical decisions with the business goals of the organization. This course examines the key aspects of software engineering economics, including life cycle economics, risk and uncertainty, analysis methods, and practical considerations which tie concept and theory to contemporary software economic realities. This course enables students to understand and apply the key software engineering economic fundamentals to real-world software economic issues.

Contents

1. The software life-cycle: waterfall model, life-cycle phase/activity
2. The software life-cycle: software work breakdown structure, software maintenance
3. The basic COCOMO model
4. Development, COCOMO effort and schedule equations
5. Phase distribution of effort and schedule, activity distribution
6. Limitations of basic COCOMO
7. Activity distribution
8. Model-product level estimates
9. Component Level Estimating Form (CLEF), using CLEF with adapted software
10. Performance models and cost effectiveness models
11. Economies of scale: discrete production functions, basic production functions
12. Diseconomies of scale, diseconomies of scale on large software projects
13. Decision criteria
14. Cost analysis and interest calculation
15. Sensitivity to interest rate or discount rate, applications to SE, and figures of merit.

Pre-requisite: Nil

Recommended Texts

1. Boehm, B. (1981). *Software engineering economics*. Englewood Cliffs. New Jersy: Prentice-Hall.

Suggested Readings

1. Boehm, B. (2000). *Software cost estimation with Cocomo II*. New York: Prentice Hall.
2. Pfleeger, S., Felicia. & Lewis, R. (2005). *Software cost estimation and sizing methods: issues, and guidelines*. Santa Monica: Rand Corp.

SEEC-6103 **Big Data Analytics** **3 (3+0)**

The proliferation of digital technologies with digital storage and recording media has created massive amounts of diverse data, which can be used for marketing and many other purposes. The concept of Big Data refers to massive and often unstructured data, on which the processing capabilities of traditional data management tools result to be inadequate. The main goal of this course is to help students learn, understand, and practice Big Data analytics and machine learning approaches. This course includes the study of Big Data technologies, machine learning techniques, and scaling up machine learning approaches with focus on industrial applications. Big Data comprises of many technologies including, but not limited to MapReduce, Apache, Pig, Sqoop, Oozie, and Spark. At the end of the course, the students will be able to understand the Big Data platforms, its use cases, and MapReduce jobs. Students will learn how to use, deploy, and maintain Apache Spark and apply analytics on structured and unstructured data with Scala.

Contents

1. Overview of big data
2. Big data in real life: the applications in marketing, analytics, retail, defense etc.
3. Introduction to Hadoop, functioning of Hadoop, cloud computing
4. Hadoop Ecosystem: HDFS, MapReduce, YARN, HBase, Hive, Pig, Sqoop, etc.
5. MapReduce framework and uses of MapReduce.
6. Databases and data warehouses
7. Data storage using HDFS
8. Collection and storage data in HDFS, processing distributed data using Apache Spark.
9. Scala Basics: variables, functions, loops, class, objects, case classes
10. Scala: tuples, map, fold, split, array concatenation, lists, iterators
11. Apache Spark: resilient distributed datasets, immutability, configurations, and context
12. Apache Spark: map partition with index, accumulators, and broadcasts
13. Running an Apache Spark job on cluster

Pre-requisite: Nil

Recommended Texts

1. Cady, F. (2017). *Handbook for Data Scientists*. Somerset: John Wiley & Sons, Inc.

Suggested Readings

1. Stephens-Davidowitz, S., & Pabon, A. (2017). *Everybody lies: Big data, new data, and what the internet can tell us about who we really are*. New York: HarperCollins.
2. Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: big data processing made simple*. CA: O'Reilly Media.

SEEC-6104 **Software CASE Tools & Applications** **3(3+0)**

CASE tools are a set of software application programs, which are used to automate Software Development Life Cycle (SDLC) activities. They are used by software project managers, analysts, and engineers to develop software systems. There are a number of CASE tools available to simplify various stages of SDLC such as analysis tools, design tools, project management tools, database management tools, documentation tools are to name a few. Use of CASE tools accelerates the development of a project to produce the desired results and helps to uncover flaws before moving ahead with the next stage in software development. This course provides an overview of software CASE tools, types, and applications. It explores topics such as case environments in practice, service-based model of CASE environment, the role of process in integrated CASE environments, and integration of CASE tools with configuration management systems. At the end of the course, students will be able to use CASE tools for data modeling, process modeling, and UI modeling.

Contents

1. Introduction to CASE and types of CASE tools
2. CASE environment, expectations about CASE, and the need for tool integration
3. Approaches to CASE tool integration and the conceptual model of integration
4. Evolution of integrated CASE environment architectures
5. Integration as a design activity
6. Service-based model of a CASE environment
7. Properties and types of integration mechanism
8. The role of process in integrated CASE environments
9. Examples of process and CASE tool interactions
10. Replacing the message service in a CASE integration framework
11. Integration of CASE tools with CM systems
12. Case environments in practice: background, observations, CASE environment progress
13. Monitoring a location and building a location tracker
14. Object-oriented analysis & design modeling
15. Comparison of popular CASE tools

Pre-requisite: Nil

Recommended Texts

1. Brown, A. W., Carney, D. J., & Morris, E. J. (1994). *Principles of case tool integration*. New York: Oxford University Press.

Suggested Readings

1. Muller, H., Norman, R. & Slonim, J. (1996). *Computer aided software engineering*. Boston: Kluwer.

SEEC-6105 **E-Commerce Applications Development** **3(3+0)**

E-commerce (electronic commerce) is the activity of electronically buying or selling products on online services or over the Internet. The purpose of this course is to introduce e-commerce applications, its impacts on business processes, and key technologies used in the development of web-based business information systems and applications. It reviews foundations of e-commerce, its infrastructure, current business models, security, web site design strategies, payment systems, and various related issues. The course covers every step of the design and building process involved in creating a powerful, extendable E-commerce website. A major part of the course will be devoted to hands-on practices covering client-side and server-side applications in web-based business information systems. Essentials of contemporary programming tools for e-commerce development such as HTML, XML, PHP, and MYSQL will be explored. Students will learn how to create and manage a product catalog, build and integrate a shopping cart, and process customer accounts and payment methods.

Contents

1. An overview of E-commerce
2. Planning an E-commerce framework
3. Products and categories
4. Product variations and user uploads
5. Enhancing the user experience
6. Creating the shopping basket and managing its content
7. The checkout and order processing
8. Shipping methods, shipping costs, shipping rules, tracking, and tax calculation
9. Discounts, vouchers, and referrals: discount codes, purchasable voucher codes, referrals
10. Checkout process consideration, order process review, authentication and confirmation
11. Taking payment for orders: payment system and gateway, online and offline payments
12. User account features: user account area, changing details, viewing & managing orders
13. Administration: managing products, categories, orders, customers, refunds, shipping, etc.
14. Deploying, security, and maintenance
15. SEO

Pre-requisite: Nil

Recommended Texts

1. Peacock, M. (2010). *PHP 5 e-commerce development: create a flexible framework in php for a powerful e-commerce solution*. Birmingham: Packt Pub.

Suggested Readings

1. Laudon, K. & Traver, C. (2017). *E-commerce 2017: business, technology, society*. Boston: Pearson.
2. Rayport, J. & Jaworski, B. (2004). *Introduction to e-commerce*. Boston, MA: McGraw-Hill Irwin MarketspaceU.
3. Bidgoli, H. (2002). *Electronic commerce: principles and practice*. San Diego: Academic Press.

SEEC-6106 **Enterprise Resource Planning Systems** **3(3+0)**

Enterprise resource planning (ERP) systems refer to a type of software that organizations use to manage day-to-day business activities such as accounting, procurement, project management, risk management, and supply chain operations. ERP systems tie together a multitude of business processes and enable the flow of data between them. ERP systems eliminate data duplication and provide data integrity with a single source of truth. Today, ERP systems are critical for managing thousands of businesses of all sizes and in all industries. This course will introduce students to ERP systems and shows how organizations use ERP systems to run their operations more efficiently and effectively. Students will learn about the critical success factors and implementation strategies that lead to ERP system success, and about the informational, knowledge, and decision-making opportunities afforded by them. The course discusses the role of ERP Systems to reduce inventories, waste, scrap, and rework and how to utilize resources efficiently. This course systematically presents several conceptual and pragmatic methodologies, tools and techniques for various phases of implementation in an enterprise.

Contents

1. Introduction to enterprise resource planning systems
2. ERP technology, ERP and business process re-engineering
3. Systems diagramming and the process map
4. ERP life cycle
5. Planning and package selection
6. Implementation, operation and maintenance
7. ERP sales
8. CRM and knowledge management
9. ERP financials
10. Human capital management, self-service, and outsourcing
11. Manufacturing systems and supply chain
12. Auditing ERP
13. Business intelligence and performance management

Pre-requisite: Nil

Recommended Texts

1. Bradford, M. (2008). *Modern ERP: select, implement & use today's advanced business systems*. Raleigh: North Carolina State University, College of Management.

Suggested Readings

1. Olson, D. (2004). *Managerial issues of enterprise resource planning systems*. Boston, MA: McGraw-Hill/Irwin.
2. Monk, E. & Wagner, B. (2009). *Concepts in enterprise resource planning.* Boston, MA: Course Technology Cengage Learning.

SEEC-6107 **Mobile Application Development** **3(3+0)**

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Mobile application development has its roots in more traditional software development with a difference that mobile apps are device-specific. Mobile apps are often written specifically to take advantage of the unique features a particular mobile device offers, like a game that uses the accelerometer of a mobile device or a health app that takes advantage of a smartwatch's heart rate sensor. This course will allow participants to gain a comprehensive understanding of all the aspects involved in developing mobile applications. The course is centered on Android, as it is a dominant mobile application development platform on the market. Topics include accessing device capabilities, industry standards, operating systems, and programming for mobile applications using an Android Studio. Upon completion, the students will be able to understand Android architecture and framework for mobile application development, explain general principles of mobile app development, and be able to create basic Android-based applications for mobile devices.

Contents

1. Introduction, installing and configuring the Android SDK manager
2. Anatomy of an Android application
3. How to get started with your first android project
4. Activities, intents, fragments, calling built-in applications using intents, displaying notifications
5. Understanding the components of a screen, adapting to display orientation, utilizing the action bar
6. Creating the user interface programmatically and listening for UI notifications
7. Using basic views, using picker views, using list view, understanding specialized fragments
8. Using image views to display pictures, using menus with views, analog and digital clock views
9. Saving and loading user preferences, creating and using databases
10. Sharing data in Android using a content provider, creating and using the content provider
11. Sending and receiving SMS messages and e-mail
12. Displaying maps, getting location data, monitoring a location
13. Creating your own services and communication between a service and an activity
14. Binding activities to services and understanding threading

Pre-requisite: Nil

Recommended Texts

1. Meier, R. (2012). *Professional android 4 application development*. Indianapolis, IN: Wiley/Wrox.

Suggested Readings

1. Conway, J. & Hillegass, A. (2012). *IOS programming: The big nerd ranch guide*. Atlanta: Big Nerd Ranch.
2. Phillips, B. & Hardy, B. (2013). *Android programming: The big nerd ranch guide*. Atlanta: Big Nerd Ranch.

SEEC-6108 **Enterprise Application Development** **3 (3+0)**

The challenge to IT professionals today is to efficiently develop and deploy distributed applications for use on both corporate intranets and over the Internet. Companies that can do this effectively will gain strategic advantage in the information economy. The Java 2 Platform, Enterprise Edition (J2EE) is a standard set of Java technologies that streamline the development, deployment, and management of enterprise applications. The course is aimed at creating robust enterprise applications using J2EE technologies that allows for rapid change and growth. The course offers the necessary skills to design, implement, configure, and deploy enterprise level desktop/web applications using Java programming language. It also provides an overview of the component, service, and communication technologies supported by the J2EE platform. Topics covered include the concepts of object-oriented programming, software architectures, challenges and platform of enterprise application development, web platform technologies, the client tier, the web tier, the enterprise information system tier, deployment, version control system, source code management, and security model of enterprise computing environments and infrastructures.

Contents

1. Object-oriented programming review
2. Software architectures: desktop, file/server, 2-tier client/server, multi-tier client/server
3. Challenges of enterprise application development and the platform for enterprise solutions
4. J2EE scenarios
5. J2EE platform technologies: component, service, and communication technologies
6. J2EE platform roles and services
7. The client-tier
8. The web-tier
9. The enterprise JavaBeans-tier
10. Integrating with the enterprise information system
11. Packaging and deployment
12. Transaction management
13. Security
14. J2EE internationalization and localization
15. Architecture of the sample application

Pre-requisite: Nil

Recommended Texts

1. Inderjeet. (2002). *Designing Enterprise applications with the J2EE platform*. Boston, MA: Addison-Wesley.

Suggested Readings

1. Sriganesh, R., Brose, G. & Silverman, M. (2006). *Mastering enterprise javaBeans 3.0*. Indianapolis, Ind: Wiley Pub.
2. Cade, M. & Sheil, H. (2010). *Sun certified enterprise architect for java ee study guide*. New York: Prentice Hall.

SEEC-6109 Artificial Intelligence 3 (3+0)

Artificial Intelligence is the ability of machines to seemingly think for themselves. AI is demonstrated when a task, formerly performed by a human and thought of as requiring the ability to learn, reason, and solve problems, can now be done by a machine. A prime example is an autonomous vehicle. The vehicle is able to perceive its surroundings and make decisions in order to safely reach its destination with no human intervention. This course will introduce the basic principles in artificial intelligence and the Prolog programming language. The course deals with a broad range of artificial intelligence topics and the students will learn representation schemes, problem solving paradigms, constraint propagation, and search strategies. The course explores different areas of AI application such as knowledge representation, natural language processing, and expert systems. At the end of this course, students will be able to apply basic principles and techniques of AI toward problem solving, inference, perception, knowledge representation, and learning.

Contents

1. Introduction to AI
2. Problem solving by searching
3. Breadth-first search, depth-first search, and depth-limited search
4. Iterative deepening, depth-first search, and comparison of uninformed search strategies
5. Informed search and exploration
6. Constraint satisfaction problems
7. Reasoning and knowledge representation
8. Inference in first-order logic
9. Introduction to Prolog programming
10. Reasoning systems for categories, semantic nets and description logics
11. Reasoning with default information
12. Reasoning with uncertainty and probabilistic reasoning
13. Representing knowledge in an uncertain domain, the semantics of Bayesian networks
14. Learning from observations: forms of learning, inductive learning, learning decision trees
15. Knowledge in learning, explanation-based learning, inductive logic programming.
16. Statistical learning, neural networks.
17. Philosophical foundations: Weak AI, Strong AI, ethics and risks of developing AI

Pre-requisite: Nil

Recommended Texts

1. Russell, S., Norvig, P. & Canny, J. (2003). *Artificial intelligence: a modern approach*. New York: Prentice Hall/Pearson Education.

Suggested Readings

1. Jones, M. (2008). *Artificial intelligence: a systems approach*. Hingham, MA: Infinity Science Press.
2. Lucci, S. & Kopec, D. (2013). *Artificial intelligence in the 21st century: a living introduction*. Dulles, VA: Mercury Learning and Information.

SEEC-6111 **Systems Programming** **3 (3+0)**

This course is intended to be a practical introductory course to design and develop system software. System software are programs written in a machine language or a low-level language on a given hardware. The objective of this course is to provide students with a basic understanding of the issues involved in writing system programs on a Windows based system. It will introduce students to the fundamental concepts and then explore related topics in detail. Topics include machine and operating system organization, process and memory management, inter-process communication, interrupts, I/O devices and ports, hardware/software interfaces, hardware-specific constraints on software applications, and using application programming interfaces and system libraries for the design and development of systems applications. Students will develop an understanding of the role of systems programming, system programming languages, and application of those languages to system level problems. By the end of the course, students will be able to design, write, and test moderately complicated low-level programs using a systems programming language.

Contents

1. Introduction to the Microsoft Windows operating system,
2. File processing, memory management, memory mapped files and DLLs
3. Process management, threads, thread synchronization, and scheduling
4. Inter-process communication
5. Input/output device drivers (usb or parallel port)
6. File system drivers and filter drivers
7. Introduction to assembly language, 80x86 families, and program layout
8. Data definitions and basic instructions
9. Unsigned arithmetic, logic, and bit operations
10. Modules, separate assembly, argument passing
11. Libraries, combining assembly and C code
12. String instructions and arrays
13. Macros and structures
14. Floating point instructions
15. Bit MS-DOS
16. Bios disk accessing
17. Bios keyboard, video/graphics
18. Interrupts, TSR programs
19. Accessing I/O ports, 8253 timers

Recommended Texts

1. Irvine, K. (2011). *Assembly language for x86 processors*. New York: Prentice Hall.

Suggested Readings

1. Hart, J. (2010). Windows system programming. New York: Addison-Wesley.

2. Baker, A. (1996). *Windows nt device driver book: a guide for programmers, with disk with Cdrom*. New York: Prentice Hall PTR.

CSEC-6112 Management Information Systems 3 (3+0)

The aim of the course is to enable students to assess the opportunities and problems that managers in a wide range of organizations face. This course also helps students to understand transformational changes within and across the industries. These changes have strategic implications for many businesses. The course provides an 'Overview' of fundamental MIS concepts, using integrated Framework for 'Decision Making' and Analyzing Information Systems. The course is comprised of different types of Information Systems available for Business use in Decision Making and Business Processes systems, covering Competitive Advantage, Executive Information Systems, and Decision Support Systems. Understand basic concepts of information technology management, its application, development themes, development methodologies, development tools and technologies of Information Systems. Able to solve common business problems and produce effective solutions to business problems. Able to design a database application to solve a business problem. Participate in an organization's information systems and technology decisionmaking processes.

Contents

1. Overview of business strategy frameworks and organizational strategies
2. Brief overview of information systems strategy
3. Understanding the organization
4. Evolution of information resources and information resources as strategic tools
5. Strategic alliances and risks.
6. Information technology and organizational design
7. Information technology and management control systems
8. Information technology and culture
9. Information technology and the design of work
10. Information technology and changing business processes
11. Information systems sourcing.
12. Using information ethically
13. Project management techniques, organizational techniques, and people techniques
14. IT project development methodologies, managing business knowledge
15. Organizational theory, management, and control
16. Types of business information system and interaction of business systems
17. Computer system concepts, computers in business, system development

Recommended Texts

1. Kroenke, D. M., Gemino, A. C., & Tingling, P. M. (2012). *Experiencing mis*. Boston, MA: Pearson.

Suggested Readings

1. Baltzan, P., Phillips, A. L., Lynch, K., & Blakey, P. (2008). *Business driven information systems*. New York, NY: McGraw-Hill/Irwin.
2. Avison, D. & Fitzgerald, G. (2006). *Information systems development: methodologies, techniques*

and tools. London: McGraw-Hill.
3. Stair, R. & Reynolds, G. (2012). *Fundamentals of information systems.* Boston, MA: Course Technology/Cengage Learning.

CSEC-6113 Game Application Development 3 (3+0)

Game development is the art of creating computer-based games involving the design, development, and release of a game. Computer games can be played today on a number of platforms. The purpose of this course is to familiarize students with the tools and practices of game development for PC and mobiles. It prepares the students for a career in software development with a particular emphasis on computer games. In this course, students will learn about the game development theory, development cycle, development team roles and responsibilities, framework, event-driven programming, project management, marketing, maintenance, and the future of game development. It will equip students with tools and techniques to develop quality games. Students will learn how to do visual character development, create a game story, apply audio, create gameplay rules, design game levels, and a compelling user interface. After completion of the course, students will be able to design and develop a complete computer game from start to finish.

Contents

1. Building the foundation and historical elements
2. Platform, player modes, and framework
3. Goals and genres
4. Player elements, player motivation, geographic, psychographics
5. Demographics, gender, generation, rating, applying player market to platform
6. Story and character development: classic charters, traditional story structure, story element
7. Plot, game story devices
8. Game characters, and point-of-view
9. Visual character development, verbal character development and, character description
10. Game storytelling
11. Game documentation
12. Gameplay: rules to play, interactivity modes, game theory, challenges, balance
13. Levels: level design, structure, time, and space
14. Interface: player-centered design, interface and its types, game features and usability
15. Audio: game audio, sound effect, voiceover, and music
16. Role & responsibilities: company role, team roles, tools, business side of game development
17. Production and management, development phases, game documentation

Recommended Texts

1. Saunders, K. & Novak, J. (2013). *Game development essentials*. Clifton Park, N.Y: Delmar, Cengage Learning.

Suggested Readings

1. Unger, K. & Novak, J. (2012). *Mobile game development*. Australia Clifton Park, N.Y: Delmar/Cengage Learning.

CSEC-6114 **Software Metrics** **3 (3+0)**

Software Metrics has become essential to good software engineering and effective project management. A software metric is a measure of software characteristics which are quantifiable or countable. Software metrics are used as a tool to help manage the development and maintenance of computer software. This course uncovers what software metrics are, why they are used, who should develop the metrics, when they should be applied, who should own the measurements, and what should be done with them. The course presents the fundamentals of measurement, classification of software measures, experimentation, data collection, and analysis. It is composed of the following basic modules: measurement theory, software product and process measurements, and measurement management. Students will be able to critically evaluate and discuss different software matrices and their applications in the real world. At the end of the course, students will be able to explain how quantitative and empirical methods are applied to software engineering problems.

Contents

1. Overview of software metrics: measurement and scope
2. The basics of measurement, measurement and models, measurement scales and scale types
3. Classifying software measures, determining what to measure
4. Applying the framework and software measurement validation
5. Empirical investigation: principles, planning experiments, and planning case studies
6. Analyzing software measurement data
7. Metrics for decision support
8. Measuring internal product attributes
9. Functional size measures
10. Estimators, applications of size measures
11. Design-level attributes, object-oriented structural attributes and measures
12. Modeling software quality
13. Measuring aspects of quality
14. Usability measures, maintainability measures, and security measures
15. Software reliability: measurements, prediction, and basics of reliability
16. Software test metrics and object-oriented metrics

Recommended Books

1. Fenton, N., & Bieman, J. (2014). *Software metrics: a rigorous and practical approach*. Boca Raton, FL: CRC press.

Suggested Books

1. Pandian, C. R. (2003). *Software metrics: A guide to planning, analysis, and application*. Boca Raton, FL: CRC Press.
2. Kan, S. (2003). *Metrics and models in software quality engineering*. Boston, MA: Addison-

Wesley.

SEEC-6115 Multimedia System and Design 3 (3+0)

Multimedia has become an indispensable part of our daily life. Multimedia is content that uses a combination of different content forms such as text, audio, images, animations, video, and interactive content. In this course, students will be introduced to principles and current technologies to generate and process multimedia, and gain hands-on experience in this area. Issues in effectively representing, processing, and retrieving multimedia data such as sound and music, graphics, and video will be addressed. The course includes topics such as image data representations, colors, digital audio, lossless compression algorithms, lossy compression algorithms, image compression standards, basic video compression techniques, and audio compression techniques. By the end of the course, students will be able to comprehend the multimedia system concepts and design principles, master the multimedia algorithms in content presentation, storage, and transmission, be familiar with multimedia standards, tools and applications, and acquire the knowledge of building multimedia systems in practice.

Contents

1. Introduction
2. Multimedia authoring and tools
3. Handling images
4. Handling sound
5. Handling animation
6. Handling video
7. Making multimedia
8. Multimedia skills
9. Planning
10. Costing
11. Designing
12. Producing
13. Content
14. Talent
15. The internet and multimedia
16. Designing for the world wide web
17. Delivering multimedia product

Recommended Texts

1. Vaughan, T. (2011). *Multimedia: making it work*. New York: McGraw-Hill.
2. Nian. & Drew, M. (2004). *Fundamentals of multimedia*. New York: Pearson Prentice Hall.

Suggested Readings

1. Chapman, N. & Chapman, J. (2004). *Digital multimedia*. Chichester: Wiley.
2. Austerberry, D. (2005). *The technology of video and audio streaming*. Burlington, MA: Focal Press.
3. Shih, F. (2013). *Multimedia security: watermarking, steganography, and forensics*. Boca Raton, FL: CRC Press.
4. Cunliffe, D. & Elliott, G. (2005). *Multimedia computing*. Colchester: Lexden Pub.

SEEC-6116 **Virtual Reality** **3 (3+0)**

Virtual reality (VR) is a powerful emerging technology that promises to change our lives unlike any other technology. By artificially stimulating our senses, our bodies become tricked into accepting another version of reality, which has a wide range of applications from training to entertainment. Virtual reality technology is evolving rapidly, making it undesirable to define VR in terms of specific devices that may be outdated after a few years. This course is concerned with fundamental principles that are less sensitive to particular technologies that keep on changing with time. At the end of the course, the students will be able to understand fundamental techniques, processes, technologies, and equipment used in virtual reality systems. Students will acquire basic knowledge about the physiology of human vision, hearing, and perception. The main objective of this course is to enable students to understand human interaction with virtual reality interfaces and the recent applications of virtual reality.

Contents

1. Definition of VR, modern experiences, historical perspective
2. Overview of VR systems: hardware, sensors, displays, software, virtual world generator
3. Human senses, perceptual psychology, and psychophysics
4. The geometry of virtual worlds: geometric modeling and transformations
5. Light propagation, lenses and images, diopters, spherical aberrations, optical distortion
6. Lens aberrations, spectral properties, the eye as an optical system, cameras, visual displays
7. Physiology of human vision
8. Visual perception: depth, motion, vection, color
9. Combining information from multiple cues and senses
10. Visual rendering: graphical rendering, ray tracing, shading, BRDFs, rasterization
11. VR rendering problems, anti-aliasing, distortion shading, and image warping
12. Motion in real and virtual worlds
13. Tracking: tracking systems
14. Estimating rotation, IMU integration, and drift errors
15. Interaction: remapping, locomotion, manipulation, social and specialized interactions
16. Audio: propagation, perception, localization, acoustic modeling, and rendering
17. Evaluating VR systems and experiences
18. Touch, haptics, taste, and smell
19. Robotic interfaces, telepresence, and brain-machine interfaces

Recommended Texts

1. Steven, M. L. (2019). *Virtual Reality*. Cambridge: Cambridge University Press.

Suggested Readings

1. Shirley, P., Marschner, S. & Ashikhmin, M. (2009). *Fundamentals of computer graphics*. Natick, MA: A K Peters.

SEEC-6117 **IOT** **3 (3+0)**

The Internet of Things (IoT) is everywhere and its explosive growth is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. IoT provides advanced data collection, connectivity, and analysis of information collected by computers everywhere by taking the concepts of Machine-to-Machine communication farther than ever before. This introductory course enables students to understand the foundations, significance, infrastructure, and architecture of IoT. Communication protocols for IoT and network, transport and application layers are discussed along with the security and privacy issues in IoT. This course also examines the topics of virtualization, Cloud computing, Fog computing, distributed data analysis, and Big Data analytics in real IoT applications. Students will also gain familiarity with associated programming frameworks, devices, and their applications. At the end of the course. students will be able to design IoT solutions and implement the basic principles of IoT with Fog and Cloud.

Contents

1. Introduction of the internet of things
2. IoT infrastructures
3. Device/cloud collaboration framework for intelligence applications IOT
4. Communication protocols for IoT and network
5. Transport and application layers
6. Fog computing
7. Programming frameworks for IoT
8. Virtualization on embedded boards as enabling technology for the cloud of things
9. Micro virtual machines (micro VMS) for cloud-assisted cyber-physical systems
10. Design and implement scalable, flexible
11. Open IoT solutions using web technologies
12. IoT data management and analytics
13. A framework for distributed data analysis for IoT
14. Security and privacy in the internet of things
15. Devices in the IoT applications
16. Cloud-based smart-facilities management

Recommended Texts

1. Buyya, R. & Dastjerdi, A. (2016). *Internet of things: principles and paradigms*. Cambridge, MA: Morgan Kaufmann is an imprint of Elsevier.

Suggested Readings

1. Somani, A. K., & Deka, G. C. (Eds.). (2017). *Big data analytics: Tools and technology for effective planning*. Boca Raton, FL: CRC Press.
2. Guinard, D. & Trifa, V. (2016). B*uilding the web of things: with examples in Node.js and Raspberry Pi*. Shelter Island, New York: Manning Publications.

SEEC-6118 **Semantic Web Techniques** **3 (3+0)**

Semantic Web is an extension of the World Wide Web (WWW) through standards by the WWW Consortium (W3C). The term was coined by Tim Berners-Lee for a web of data that can be processed by machines. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The aim of this course is to teach the students the concepts, technologies and techniques underlying and making up the Semantic Web. In this course students will be introduced to the languages and tools useful in Semantic Web programming. They will understand how this technology revolutionizes the WWW and its uses. Ontology languages and technologies will be covered. In addition, students will be exposed to ontology engineering, application scenarios, semantic web query languages, and semantic web applications. At the end of the course the students should be able to understand and discuss fundamental concepts, advantages and limits of the semantic web, and relationship between Semantic Web and Web 2.0. Furthermore, they will be able to understand and use ontologies, use the RDF framework, and associated technologies.

Contents

1. Introduction to the semantic web and ontologies
2. Ontology languages for the semantic web
3. Resource Description Framework (RDF)
4. Lightweight ontologies
5. RDF Schema
6. Web Ontology Language (OWL)
7. Query language for RDF
8. SPARQL
9. Ontology Engineering
10. Semantic web and Web 2.0
11. Applications of semantic web

Recommended Texts

1. Yu, L. (2007). *Introduction to the semantic web and semantic web services*. Boca Raton, FL: CRC Press.
2. Segaran, T., Evans, C., & Taylor, J. (2009). *Programming the Semantic Web: Build Flexible Applications with Graph Data*. Sebastopol, CA: O'Reilly.

Suggested Readings

1. Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). *Foundations of semantic web technologies*. Boca Raton, FL: CRC press.

SEEC-6119 **Mobile Computing** **3 (3+0)**

With the increasing popularity of mobile devices, mobile computing has become part of our daily life. To design software that runs on a mobile device the limitations of different mobile devices like limited memory, performance, storage, and different operating systems must be considered. This course provides guidelines, design principles, and experience in developing applications for mobile devices. An overview of programming mobile devices is provided and major issues in programming mobile devices are discussed. Topics such as memory management, the concepts of applications, dynamically linked libraries, concurrency, handling local resources, networking, and security features are covered. During this course, students will be able to grasp the concepts and features of mobile computing technologies and applications. At the end of the course, students will have a working understanding of the characteristics and limitations of mobile hardware devices, the ability to develop applications that are mobile-device specific, and will be aware of issues relating to security and privacy of user data.

Contents

1. Introduction to mobile computing and the architecture of mobile software applications
2. Creating consumable web services for mobile devices
3. Memory management
4. Mobile applications
5. Mobile user interface design
6. Dynamic linking
7. Concurrency
8. Managing resources
9. Security
10. Introduction to mobile application development with Android
11. Introduction to mobile application development with iOS
12. Introduction to mobile application development with Windows phone
13. Introduction to mobile application development with Blackberry

Recommended Texts

1. Mikkonen, T. (2007). Programming mobile devices: an introduction for practitioners. Chichester, England Hoboken, NJ: John Wiley.

Suggested Readings

1. McWherter, J. & Gowell, S. (2012). *Professional mobile application development*. Indianapolis, Ind: John Wiley & Sons, Inc.
2. Far, R. (2005). *Mobile computing principles: designing and developing mobile applications with UML and XML*. Cambridge England New York: Cambridge University Press.
3. Fling, B. (2009). *Mobile design and development*. Beijing Sebastopol, Calif: O'Reilly.
4. Adelstein, F. (2005). *Fundamentals of mobile and pervasive computing*. New York, NY: McGraw-Hill.

SEEC-6120 **Data Warehousing** **3 (3+0)**

The course introduces the concepts of database technology used for decision-support infrastructure and analytics. A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources. In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users. The objective of this course is to develop the concepts and skills required for designing and implementing data warehouses. Students will learn how to create data integration workflows and use different tools for data warehousing. By the end of the course, students will have the design experience, software background, and organizational context that prepares them to succeed with data warehouse development projects.

Contents

1. Introduction to data warehousing: characteristics, architecture, and data staging
2. Multidimensional model, meta-data, accessing data warehouse
3. ROLAP, MOLAP, and HOLAP
4. Data warehouse system lifecycle: risk factors, top-down vs bottom-up
5. Data mart design phases and methodological framework
6. Data-driven, requirement-driven, testing data marts
7. Analysis and reconciliation of data sources
8. User requirement analysis
9. Conceptual modeling: dimensional fact model, events and aggregation, temporal aspects
10. Overlapping fact shcemata, formalizing the dimensional fact model
11. Conceptual design: ER, relational, and XML schema-based design, mixed-approach design
12. Requirement-driven approach design

Recommended Texts

1. Golfarelli, M. & Rizzi, S. (2009). *Data warehouse design: modern principles and methodologies*. New York, NY: McGraw-Hill.

Suggested Readings

1. Inmon, W. (2005). *Building the data warehouse*. Indianapolis, IN: Wiley.
2. Kimball, R. (1998). *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses*. New York: Wiley.
3. Ponniah, P. (2010). *Data warehousing fundamentals for IT professionals*. Hoboken, NJ: John Wiley & Sons.

SEEC-6121 **Data Mining** **3 (3+0)**

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. It is currently regarded as the key elements of a more general process called Knowledge Discovery that deals with extracting useful knowledge from raw data. The knowledge discovery process includes data selection, cleaning, coding, using different statistical and machine learning techniques, and visualization of the generated structures. Students will be introduced to the principles and techniques involved in data mining such as statistical methods, decision trees, artificial neural networks, cluster analysis, and genetic algorithms. Important related technologies of data warehousing and on-line analytical processing (OLAP) will be also discussed. At the end of the course, the students will be able to understand the applications, concepts, and techniques of data mining. Learn different data mining tools and apply basic data mining techniques to actual problems.

Contents

1. Data-mining concepts: process, large data sets, data warehouses, business aspects
2. Preparing the data
3. Data reduction
4. Learning from data
5. Statistical methods
6. Decision trees and decision rules
7. Artificial neural networks
8. Ensemble learning
9. Cluster analysis
10. Association rules
11. Web mining and text mining
12. Genetic algorithms
13. Fuzzy sets and fuzzy logic
14. Visualization methods
15. Data mining tools: Weka, CBA and Yale, etc.

Recommended Texts

1. Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. Piscataway, New Jersey Hoboken, NJ: IEEE Press Wiley.
2. Han, J., Kamber, M. & Pei, J. (2012). *Data mining: concepts and techniques*. Amsterdam Boston: Elsevier/Morgan Kaufmann.

Suggested Readings

1. Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT

Press.
2. Singh, R. & Asthana, A. (2012). *Data Mining and Data Warehousing Practical Machine Learning Tools Techniques.* Saarbrücken: LAP LAMBERT Academic Publishing.

SEEC-6122 Business Intelligence and Analytics 3 (3+0)

Business Intelligence (BI) is the field of combining data, technology, business processes, and analytics to optimize business decisions and drive success. The ability to generate insights using data in today's world is crucial for any organization's success into the future. Turning data into valuable information is a necessity for ever-changing markets. This course will give an introduction to analytical tools and skills that can be used to understand, analyze, and evaluate the challenges and opportunities for an organization. Students will learn about various technical aspects of BI and understand the processes involving in planning, designing, building, and maintaining the BI environment. Further, this course will develop skills in the use and application of various techniques and tools for driving insights from data for effective business decision making.

Contents

1. Value drivers, performance metrics and key performance indicators
2. Use cases for BI
3. BI success factors, strategic versus tactical planning, BI strategy and plan
4. BI environment: analytics platform, frameworks, services, and systems evaluation
5. Business process and information flow
6. Data requirements analysis
7. Data warehouses and the technical BI architecture
8. Data profiling
9. Business rules
10. Data quality
11. Data integration
12. Deriving insight from data
13. Knowledge discovery & delivery
14. Installations, configuring and maintaining the BI server
15. Creating reports using answers and dashboards.

Recommended Texts

1. Rittman, M. (2013). *Oracle business intelligence 11g developer's guide*. New York, NY: McGraw-Hill.

Suggested Readings

1. Larson, B. (2012). *Delivering business intelligence with Microsoft SQL server 2012*. New York, NY: McGraw-Hill.
2. Vitt, E., Luckevich, M. & Misner, S. (2002). *Business intelligence: making better decisions faster*. Redmond, Wash: Microsoft Press.

3. Laberge, R. (2011). *The data warehouse mentor: practical data warehouse and business intelligence insights*. New York, NY: McGraw-Hill.

SEEC-6123 Database Administration & Management 3 (3+0)

Every organization that manages data using a database management system (DBMS) requires one or more database administrators to ensure the effective use and deployment of its databases. The course of Database Administration & Management is designed for students to gain a comprehensive overview of all the skills necessary to become successful database administrators. This course provides an understanding of the internal functionality and features of the database management system by Oracle corporation. Students are required to have a working knowledge of the relational database model as well as PL/SQL programming skills. Students will learn how to administrator databases implemented in Oracle DBMS. They will learn how to install and configure DBMS, perform common administrative tasks using GUI, and know how to use the SQL language to manipulate data and DBMS. In addition to practicing existing recovery, backup, and network plans, students will gain hands-on experience practicing database-related procedures such as how to monitor a database, statistics, usage, and manage the performance of the database.

Contents

1. Oracle installation and Oracle architectural components
2. Oracle physical structures
3. Managing an Oracle Instance
4. Creating database and data dictionary
5. Managing control files and redo log files
6. Managing tablespaces, data files, segments, and blocks
7. Managing undo data, indexes and privileges
8. Maintaining data integrity and constraints
9. Basic Oracle Net Architecture
10. Server-side configuration
11. Client-side configuration
12. Backup and recovery
13. Rollback segments
14. Latches, rollback, shared servers, locks, and block efficiency
15. Statistics and monitoring index usage

Recommended Texts

1. Bryla, B. & Loney, K. (2008). *Oracle database 11g DBA handbook*. New York, NY: McGraw-Hill.
2. Mullins, C. (2013). *Database administration: the complete guide to DBA practices and procedures*. New York: Addison-Wesley.

Suggested Readings

1. Connolly, T. & Begg, C. (2010). *Database systems: a practical approach to design, implementation, and management*. Boston, MA: Addison-Wesley.
2. Loney, K. (2009). *Oracle database 11g: the complete reference*. New York, NY: McGraw-Hill.

SEEC-6124 **Advance Database Management** **3(3+0)**

This course will address the advanced issues in modern database systems and applications. Databases underlie most complex computing systems. Software systems that involve large and complex databases are heterogeneous. Simple approaches designed for small, centralized, homogeneous databases are ineffective and inappropriate for dealing with large, distributed, heterogeneous environments. Data-related issues in building, analyzing, and maintaining complex software systems are the focus of this course. Techniques for operating and maintaining heterogeneous database systems, business problem-solving, and decision-support systems are also discussed. At the end of the course, the students will be able to understand the concepts of transaction processing, concurrency control, crash recovery, and database security along with their practical application. Furthermore, this course also provides an overview of data warehousing, OLAP, and data mining.

Contents

1. PL/SQL: introduction to PL
2. Transaction-processing monitors
3. Transactional workflows
4. Main-memory databases
5. Real-time transaction systems and long-duration transactions
6. Transaction management in multi-databases
7. Concurrency control: locks
8. Optimistic concurrency control and timestamping concurrency control.
9. Object-based databases and xml object-based databases
10. Programming languages: OO vs OR
11. XML: structure, document schema, querying, API, and applications
12. Data warehousing and its design
13. OLAP and data mining
14. Database security

Recommended Texts

1. Silberschatz, A., Korth, H. & Sudarshan, S. (2011). *Database system concepts*. New York, NY: McGraw-Hill.

Suggested Readings

1. Connolly, T. & Begg, C. (2010). *Database systems: a practical approach to design, implementation, and management*. Boston: Addison-Wesley.

2. Bayross, I. (2002). *SQL, Pl/SQL the programming language of Oracle*. New Delhi: BPB Publications.
3. Han, J., Kamber, M. & Pei, J. (2012). *Data mining: concepts and techniques*. Amsterdam Boston: Elsevier/Morgan Kaufmann.
4. Gertz, M. & Jajodia, S. (2008). *Handbook of database security: applications and trends*. New York: Springer.

SEEC-6125 Business Process Management 3 (3+0)

This course looks at ways in which business processes can be analyzed, redesigned, and improved thus ensuring that they are meeting the needs of customers and the enterprise. A business process is a set of related activities that together realize a business goal in an organizational and technical context. These processes take place in a single organization but may need to interact with processes in other organizations. This course will introduce you to business process management. You'll learn how business processes can help you improve your company's bottom line by providing a higher level of quality and consistency for your customers. Business Process Management (BPM) is concerned with the concepts, methods, and techniques that support the design, improvement, management, configuration, enactment, and analysis of business processes that deliver lean and customer focused business processes. BPM includes process modelling that includes defining, analyzing, and improving processes. Students will be able to understand business process from a management and process analyst perspective, learn skills, analytical frameworks and general principles for managing business processes.

Contents

1. Business introduction
2. Evolution of enterprise systems architectures
3. Business process modeling
4. Process orchestrations
5. Process choreographies
6. Properties of business processes
7. Business process management architectures
8. Business process management methodology

Recommended Texts

1. Weske, M. (2012). *Business process management*: concepts, languages, architectures. Berlin New York: Springer.

Suggested Readings

1. Yvonne, L. A., Martin, B., Tony, B., Bruce, D. D., Jason, F., Daniel, J. M., & Robyn, L. R. (2009). *Business Process Management Common Body of Knowledge*. Chicago: Association of Business Process Management Professionals.
2. Becker, J., Kugeler, M. & Rosemann, M. (2011). *Process management: a guide for the design of business processes*. Berlin London: Springer.

3. Jeston, J. & Nelis, J. (2008). *Business process management: practical guidelines to successful implementations.* Amsterdam: Elsevier/Butterworth-Heinemann.
4. Malik, T. (2009). *Process management: practical guidelines to successful implementations*. New Delhi: Global India Publications.

SEEC-6126 Knowledge Management 3 (3+0)

Knowledge, knowledge products, and knowledge processes are the key ingredients of productivity and profitably in the business world. Knowledge revolution has given birth to knowledge economies. Knowledge management (KM) is the field of managing human knowledge. Knowledge Management is commonly associated with processes like knowledge creation, knowledge sharing, knowledge storage, knowledge refinement, etc. Knowledge management (KM) is an area that has captured the attention of many organizations that are concerned with the way's knowledge is managed more effectively. KM offers systematic methods in leveraging and managing organizational knowledge through KM processes of creation, storing, sharing, and application of knowledge. The need is to gain a sustainable competitive edge among partners as well as competitors by learning how to leverage intangible assets in new and creative ways. This course introduces students to appraise current thoughts on knowledge management in the light of contemporary debates on knowledge productivity, strategic capability, and organizational learning. Further, it enables them to learn how to develop, manage, and evaluate knowledge management theories, models, frameworks, systems, initiatives, and best practices.

Contents

1. History and paradigms of knowledge management
2. Types of knowledge: explicit knowledge, tacit knowledge, embedded knowledge
3. KM processes: knowledge discovery/ detection
4. KM frameworks and models: SECI, Alen Frost's, and Van Buren's models
5. Knowledge capture and codification: group, knowledge codification
6. Knowledge sharing and communities of practice: types of communities
7. Knowledge application: task analysis and modeling, knowledge reuse
8. The role of organizational culture: different types of cultures
9. Knowledge management tools: knowledge blogs, mashups, PKM
10. Knowledge management strategy: knowledge audit, gap analysis
11. The value of knowledge management: ROI and metrics
12. Organizational learning and organizational memory
13. Major categories of knowledge management roles
14. The profession and ethics of knowledge management

Recommended Texts

1. Dalkir, K. (2017). *Knowledge management in theory and practice*. Cambridge, MA: Massachusetts Institute of Technology.

2. Pasha, M. & Pasha, S. (2012). *Essentials of knowledge management: concepts, theories and practices.* Innovators Knowledge Services.

Suggested Readings

1. Tiwana, A. (2007). *The knowledge management toolkit: orchestrating It, strategy, and knowledge platforms*. New York: Prentice Hall PTR.
2. Geisler, E. & Wickramasinghe, N. (2009). *Principles of knowledge management: theory, practices, and cases*. Armonk, N.Y: M.E. Sharpe.
3. Jennex, M. (2008). *Knowledge management: concepts, methodologies, tools and applications.* Hershey PA: Information Science Reference.

SEEC-6127 Natural Language Processing 3 (3+0)

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes, Context-free parsing, Earley algorithm, Extending CFG.
4. Probabilistic parsing, Parsing tricks, Human sentence processing.
5. Semantics, Forward-backward algorithm
6. Expectation Maximization.
7. Finite-state algebra, Finite-state implementation, Finite-state tagging, Noisy channels and FSTs, More FST examples.
8. Programming with regexps
9. Morphology and phonology.
10. Optimal paths in graphs
11. Structured prediction.
12. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing* (3rd ed.). New York: MIT press
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. New York: MIT press

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.

SEEC-6128 **Grammar Engineering** **3 (3+0)**

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview
3. LFG Basics.
4. LFG Basics II, Templates I, MacOSX, Unix.
5. Templates II, f-descriptions
6. Subject-Verb Agreement, Determiners, xlerc file
7. Lexical Rules, Passive and Argument alternations.
8. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
9. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
10. Generation & Optimality Projection, Restricting Over-generation
11. Complements
12. xcomp and comp.
13. Functional Uncertainty, Imperatives and empty categories.
14. Finite-State Morphology (FSM) I.
15. FSM II (-unknown), Free Word Order and the Shuffle Operator.
16. Meta-categories, Meta-rule-macros and Coordination.
17. Project

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. Stanford: CSLI. Publications.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar*. Netherlands: Brill

SEEC-6129 **Text Processing & Data Analysis** **3 (3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing
6. Text Processing Functions.
7. Loop Control, Hashes, DBM Databases
8. Advanced Sorting.
9. Regular Expressions, Environment Variables
10. CGI-Programming.
11. Process Management, References and Data Structures.
12. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. California: O'Reilly Media.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming perl: unmatched power for text processing and scripting*. California: O'Reilly Media.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl cookbook: solutions & examples for pearl programmers*. California :O'Reilly Media.
2. Lidie, S., & Walsh, N. (2002). *Mastering perl: graphical user interfaces in perl*. California: O'Reilly Media.

SEEC-6130 **Speech Recognition & Synthesis** **3 (3+0)**

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background: N-gram Language Modeling
2. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization, Phonetics Speech Synthesis, pages 1-10 , Optional Advanced Reading, Text Segmentation and Organisation, Text Decoding.
3. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software, Prosody Prediction from Text.
4. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis), Unit Selection Synthesis, Optional Advanced Reading.
5. ASR: Noisy Channel Model, Bayes, HMMs, Forward, Viterbi, Hidden Markov Models, Automatic Speech Recognition.
6. ASR: Feature Extraction and Acoustic Modeling, Evaluation, Speech Recognition: Advanced Topics.
7. ASR: Learning (Baum-Welch) and Disfluencies, Automatic Speech Recognition, Speech Recognition: Advanced Topics.

Recommended Texts

1. Jurafsky, D., Martin, J. H. (2019). *Speech and language processing* (3rd ed.). New Delhi: Pearson Education India.

Suggested Readings

1. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

MS COMPUTER SCIENCE

CSCC-7401 Advanced Theory of Computation 3 (3+0)

The course aims to introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability. Several mathematical models of computation have been formulated independently and under any such computational model, the existence of well-defined but unsolvable problems can be formally shown. These topics form part of the core of the mathematical foundations of computer science that will provide students and researchers with a sound theoretical view of the most fundamental concepts of computation. Specifically, this course provides a rigorous introduction to the theoretical foundations of computer science. It deals with a number of interconnected topics and tries to answer the basic questions, "What is a computer?", "What is an algorithm?", and "What is computable?". This course examines important theorems and proofs, establishes a number of interesting assertions in order to expose the techniques used in the area of theory of computation. Note that although this is not a "mathematics" course, it does make significant use of mathematical structures, abstractions, definitions, theorems, proofs, lemmas, corollaries, logical reasoning, inductive proofs, and the like. If such concepts are difficult for you, you will find this course very difficult but rewarding.

Contents

1. Mathematical Tools and Techniques: Logic and Proofs, Sets, Functions and Equivalence Relations, Languages.
2. Proofs, Principle of Mathematical Induction, Recursive Definitions, Structural Induction.
3. Regular Languages and Regular Expression.
4. Nondeterministic Finite Automata with A-Transitions, Kleene’s Theorem.
5. Regular, Non-regular Languages, Criteria for Regularity, Minimal Finite Automata
6. Context-Free Grammars, Regular Grammars, Derivation Tree and Ambiguity
7. Pushdown Automata: Definitions and Examples, Deterministic Pushdown Automata
8. Context-Free, Non-Context-Free Languages.
9. Turing Machines: Definitions and Examples, Non-deterministic TM, Universal TM
10. Recursively Enumerable and Languages: Recursively Enumerable and Recursive, Enumerating a Language, More General Grammars, Context-Sensitive Languages and the Chomsky Hierarchy
11. Unsolvable Problems: A No-recursive Language and an Unsolvable Problem, The Halting Problem, Unsolvable Problems Involving TMs, Rice’s Theorem, Post’s Correspondence Problem
12. Computable Functions: Primitive Recursive Functions, PRF and Bounded Operations, Unbounded. Minimalization and ų-Recursive Functions, Godel Numbering

Recommended Texts

1. John Martin. (2010). *Introduction to languages and the theory of computation*, (4^{th} ed.). New York: McGraw-Hill Science/Engineering/Math.
2. Tourlakis, George J. (2012). *Theory of computation*. Hoboken: Wiley.

Suggested Readings

1. Goddard, W. (2008). *Introducing the theory of computation*. Burlington: Jones and Bartlett Publishers, Inc.

2. Dexter C. Kozen. (2010). *Theory of computation*, (1st ed.). New York: Springer.

CSCC-7402 Advanced Algorithm and Analysis 3 (3+0)

Advanced Analysis and Algorithms is an advanced course in design and analysis of algorithms covering topics typically not covered in undergraduate algorithms. It includes introduction of formal techniques and the underlying mathematical theory. NP-completeness. Search Techniques. Randomized Algorithms. Heuristic and Approximation Algorithms. Topics include asymptotic analysis of upper and average complexity bounds using big-O, little-o, and theta notation. Fundamental algorithmic strategies (brute-force, greedy, divide-and-conquer, backtracking, branch-and-bound, pattern matching, and numerical approximations) are covered. Also, include are standard graph and tree algorithms. Additional topics include standard complexity classes, time and space tradeoffs in algorithms, using recurrence relations to analyze recursive algorithms, non-computable functions, the halting problem, and the implications of non-computability. Algorithmic animation is used to reinforce theoretical results. Upon completion of the course, students should be able to explain the mathematical concepts used in describing the complexity of an algorithm, and select and apply algorithms appropriate to a particular situation.

Contents

1. Dynamic Programming, elements, rod cutting, Longest Common Subsequence, Optimal Binary Search Trees.
2. Greedy Algorithms and methods.
3. Amortized Analysis: Aggregate Analysis, Accounting Method, Potential Method, Dynamic Tables.
4. Fibonacci Heaps: Structure of Fibonacci Heaps, Mergeable-Heap Operations.
5. VanEmde Boas Trees: Preliminary Approaches, A Recursive Structure, The Van Emde Boas Tree.
6. Dijkstra algorithm, bellmen ford algorithm and proof of shortest paths.
7. All-Pairs Shortest Paths: Shortest Paths and Matrix Multiplication, The Floyd-Warshall Algorithm, Johnson’s Algorithm for Sparse Graphs.
8. Maximum Flow: Flow Networks, The Ford-Fulkerson Method, Maximum Bipartite Matching, Push-Relabel Algorithms, The Relabel-To-Front Algorithm.
9. Multithreaded Algorithms: The Basics of Dynamic Multithreading
10. String Matching: The Naive String-Matching Algorithm
11. Approximation Algorithms: The Vertex-Cover Problem, The Traveling-Salesman Problem
12. Parallel Algorithms: Parallelism, The PRAM Model, Simple Parallel Operations.

Pre-Requisite: Analysis of Algorithms

Recommended Texts

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Cambridge: MIT press.

Suggested Readings

1. McConnell, J. J. (2001) Analysis of algorithms: An active learning approach, (2nd ed.).

Burlington: Jones and Bartlett Publishers.
2. Vazirani, V. V. (2013). Approximation algorithms. New York: Springer Science & Business Media.

CSCC-7403 Advanced Operating System 3 (3+0)

Advanced Operating Systems course addresses a broad range of topics in operating system design and implementation, including: Operating system structuring, Synchronization, communication and scheduling in parallel systems, Distributed systems, their communication mechanisms, distributed objects and middleware, Failures and recovery management, CPU scheduling: Scheduling policies and algorithms, Scheduling algorithm comparison, Real-time and multi-processor scheduling, Linux case study. Dynamic memory management: Internal design alternatives for malloc and free routines, Garbage collection. OS memory management: Memory protection, Program relocation, Memory partitioning techniques, Virtual memory, Paging and segmentation, TLB and cache management. File systems: Naming issues, Design alternatives for file systems, Example file systems and their comparison. I/O management: Memory mapped Vs Direct I/O, Interrupt driven Vs Polled I/O, Device controllers and device drivers, Naming issues, OS I/O architecture, Buffering techniques, Disk devices and their management. At the end of the course, students are expected to be proficient in details of operating systems and be sensitive to implementation and performance tuning of operating systems in preparation to entering the industry or in pursuit of graduate studies.

Contents

1. Introduction: Operating System Architecture, Multitasking, Multiuser, Multiprocessing
2. Process Scheduling Concepts, System Calls for Process Management, Process Communication and Synchronization Concepts, Memory and I/O Management Overview
3. Multitasking OS: Design and Implementation Kernel of Multitasking OS: Services, Process State Transitions, Functional Specification, Implementation Considerations, System List.
4. Multiprocessor Systems: Introduction, Parallel Hardware and Interconnections, Types of Multiprocessor OS, Sharing OS, Mutiprocesor OS Design Considerations, Threads, Thread Scheduling, Kernel Mode Processes, Multiprocessor Synchronization.
5. Memory Management: Overview, Pages, Zones, Kmalloc, Vmalloc, Slab Layer, Slab Layer Allocator, Deallocator, Statically Allocating on The Stack, High Memory Mapping.
6. Unit V I/O Systems: I/O Device Types, I/O Structure, Driver Interfaces, Disk Device Driver Access Strategies, Unification of Files And I/O Devices, Generalized Disk Device Drivers, Disk Caching
7. File Descriptors, File Blocks Allocation, Mapping of File Blocks, System Calls for The File System: Open, Read, Write, Lseek, Close. Mounting and Un Mounting File Systems, Link

Recommended Texts

1. Stallings, W. (2012). *Operating systems: internals and design principles*. Boston: Prentice Hall.
2. Tanenbaum, A. S., & Woodhull, A. S. (2006). *Operating systems design and implementation*, (3rd ed). Boston: Prentice Hall.

Suggested Readings

1. Maurice J. Bach. (1986). The design of the UNIX Operating System, Boston: Prentice Hall.
2. Andrew S. Tanenbaum. (2007). Modern operating systems, (3rd ed.). Boston: Prentice Hall.
3. Stevens, W. R., Rudoff, A. M., & Fenner, B. (2003). UNIX network programming volume 1: The

Sockets Networking API (vol. 3). Boston: Addison-Wesley Professional.
4. Love, R. (2010). Linux kernel development. New York: Pearson Education.

CSCC-7404 Advanced Computer Architecture 3(3+0)

To develop an understanding of high-performance computer architecture, as a foundation for advanced work in computer architecture. The students will get overview of computer architecture, which stresses the underlying design principles and the impact of these principles on computer performance. More specifically, Distinction between Computer Architecture, Organization and design, General operation of a stored program digital computer, The Fetch – Execute process, Concept of an ISA, Instruction set features, Addressing Modes, RISC and CISC architectures, Measures of performance, Implementing Register Transfers using Digital Logic Circuits, The Design Process, A Uni-Bus implementation for the SRC, Pipelining, Microprogramming. General topics include design methodology, processor design, control design, memory organization, system organization, and parallel processing. At the end, a student would be able to know the classes of computers, and new trends and developments in computer architecture, understand pipelining, instruction set architectures, memory addressing, understand the performance metrics of microprocessors, memory, networks, and disks, understand the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges etc

Contents

1. Introduction to Advanced Computer: Flynn's Taxonomy of Computer Architecture, SIMD, MIMD, Interconnection Networks.
2. Multiprocessors Interconnection Networks: Interconnection Networks Taxonomy, Bus-Based Dynamic Interconnection Networks.
3. Performance Analysis of Multiprocessor Architecture: Computational Models, An Argument for Parallel Architectures, Interconnection Networks Performance Issues.
4. Shared Memory Architecture: Classification of Shared Memory Systems, Bus-Based Symmetric Multiprocessors.
5. Message Passing Architecture: Introduction to Message Passing, Routing in Message Passing Networks, Switching Mechanisms in Message Passing.
6. Abstract Models: The PRAM Model and Its Variations, Simulating Multiple Accesses on an EREW PRAM, Analysis of Parallel Algorithms.
7. Network Computing: Computer Networks Basics, Client/Server Systems, Clusters
8. Parallel Programming in the Parallel Virtual Machine: PVM Environment and Application Structure.
9. Message Passing Interface (MPI): Communicators, Virtual Topologies, Task Communication.
10. Scheduling and Task Allocation: The Scheduling Problem, Scheduling DAGs

Pre-Requisites: Computer Architecture

Recommended Books

1. El-Rewini, H., & Abd-El-Barr, M. (2005). *Advanced computer architecture and parallel processing* (Vol. 42). Hoboken: John Wiley & Sons.

Suggested Readings

1. Shiva, S. G. (2018). *Advanced computer architectures*. Boca Raton: CRC Press.
2. Amit Kumar Mishra, S K Kataria & Sons. (2010). *Advanced Computer Architecture*, New Delhi: S K Kataria & Sons.
3. DezsoSima. (1997). *Advanced computer architecture: A design space approach*, Boston: Addison-Wesley.
4. Hwang, K., & Xu, Z. (1998). *Scalable parallel computing: technology, architecture, programming.* New York: McGraw-Hill, Inc.

CSEC-7401 Advanced Topics in Wireless Network 3(3+0)

This course will cover the fundamental aspects of wireless networks, with emphasis on current and next-generation wireless networks. This will cover the latest research in the area of wireless networking. These types of networks have been growing exponentially in the past several years. The advent of wireless and wired networks convergence and as the Internet is increasingly becoming the tool for a wide range of technical, economical and industrial applications, resources management becomes very crucial and vital issue for any future networks. The objectives of this course are to focus on resource management and performance analysis in transporting multimedia traffic in wireless communication networks. Topics include: traffic characteristics, connection admission control, packet scheduling, access control, and mobility and handoff management. The class will build understanding of all layers of wireless networking and the interactions between them. We will introduce the students to wireless networking research and guide them to investigate novel ideas in the area via semester-long research projects.

Contents

1. Introduction to Radio Resource Management for Wireless Networks
2. Resource Management for Circuit-Switched Services
3. Traffic Modelling
4. Access Control and Admission Control
5. Mobility Management and Handoff Management
6. MAC and Packet Transmission Scheduling
7. Resource Management in Ad Hoc Networks
8. Applications: Resource Management in Packet Access
9. CDMA2000 & WCDMA (i.e., UMTS) Systems
10. Qos and QoE
11. Internet of Things (IOT)
12. WRAN
13. CRAN,
14. Vehicular Adhoc Network

Pre-Requisites: Computer Networks

Recommended Texts:

1. J. Zander and S.-L. Kim, Radio. (2001). *Resource management for wireless networks*, Norwood: Artech House Publishers.
2. S. Kyriazakos and G. Karesos. (2004). *Practical radio resource management in wireless networks*. Boston: Artech House

Suggested Readings:

1. T. Janevski. (2003). *Traffic analysis and design of wireless IP networks*. Jamalipour, The Wireless Mobile Internet, Hoboken: Wiley.
2. M. Hassan and R. Jain, (2004). *High performance TCP/IP networking*. New Jersey: Prentice Hall.

CSEC-7401 **Advanced Artificial Intelligence** **3 (3+0)**

This course will introduce the key principles in artificial intelligence. It will cover simple representation schemes, problem solving paradigms, constraint propagation, and search strategies. Areas of application such as knowledge representation, natural language processing, expert systems, vision and robotics will be explored. The Prolog programming language will also be introduced. Upon successful completion of this course, student will be able to learn and apply searching techniques, learning techniques, identify the major approaches to AI, explain the differences between various types of logic and basic statistical tools used in AI, list the most common methods of statistical learning and classification and explain the basic differences between them, describe the components of Turing machine. At the end, student would be able to: developing their ability to survey existing research and read research papers critically; developing their ability to devise and run experiments, and analyze their results; refining their communication skills; in particular, the ability to present research work to an audience, and to discuss technical material with peers

Contents

1. Introduction: Turing Test, Strong AI vs Weak AI, Heuristics, Applications and Methods, History of AI.
2. Uninformed Search: Search in IS, Genrate-and-Test Paradigm, Blind Search Algorithm.
3. Informed Search: Heuristics, the Best-First Search, The Beam Search, The A* Search, The Bidirectional Search.
4. Search Using Game: Game Trees and Minimum, Game Theory.
5. Logic in AI: Logic and Representation, Propositional Logic, Predicate Logic, Other Logics.
6. Knowledge Representation: Search Tree, Production System, Objects, Frames, Scripts & the Conceptual Dependency System, Semantic Networks, Recent Approaches, Agents.
7. Prolog Programming.
8. Production Systems: Strong Methods vs. Weak Methods, Production System and Inference Methods, Stochastic Processes and Markov Chain.
9. Uncertainty in AI: Fuzzy Sets, Fuzzy Logic, Fuzzy Inference, Probability Theory and Uncertainty.
10. Expert Systems: Characteristics of ES, Knowledge Engineering, Knowledge Acquisition, Classical ES, Case-Based Reasoning.
11. Neural Networks: Introduction, The Perceptron Learning Rule, Back propagation, Discrete Hopfield Networks, Application Areas.
12. Evolutionary Computation: Simulated Annealing, Genetic Algorithms.
13. Natural Language Processing: History of NLP, Syntax and Formal Grammars, Syntax and Formal Grammar, Statistical Parsing, Hidden Markov Model, Wordnet, Question Answering System.
14. Automated Planning: Planning Terminology, Planning as Search, Hierarchical Planning

Recommended Texts

1. Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q., & Lee, I. *(2018). Artificial intelligence in the 21st century.* Piscataway: *IEEE* Access.
2. Russell, S., & Norvig, P. (2002). *Artificial intelligence: a modern approach*. New Jersey: Prentice Hall.

Suggested Readings

1. M. Tim Jones. (2008). *Artificial Intelligence: a systems approach*, (1st ed). Burlington: Jones and

Bartlett Publishers, Inc.

CSEC-7402 Advanced Human Computer Interaction 3 (3+0)

The aim of the course Advanced Human Computer Interaction is to identify the important research issues, and to ascertain potentially fruitful future research directions in the relation to the multimodal emotion analysis and to human-computer interaction. Upon successful completion students will be able to explain and reflect critically with the use of course literature on the idea of organizational usability and UX, outline a plan for how to use HCI to create value for and empower the employee/customer/citizen, apply findings from empirical work on HCI issues in the use of multiple organization-wide systems, describe and explain an issue with legacy systems (old systems) using literature from the course and develop and present a comprehensively set of documented and motivated prototypes, sketches, templates, running systems, or scripts that supports HCI interactions within or across organizations, and explain the principle idea behind. Recent research topics will also be discussed in the course.

Contents

1. Overview of advanced topics of human-computer interaction,
2. Design thinking and the basic practices of interaction design.
3. Importance of human-computer interaction/interface design
4. Iterative design
5. Input/output techniques
6. How to design and evaluate interfaces
7. Research topics.

Pre-Requisites: Human Computer Interaction

Recommended Texts

1. Henderson, A. (2002). *Interaction design: beyond human-computer interaction*. London: Ubiquity.
2. Robertson, J., & Kaptein, M. (Eds.). (2016). *Modern statistical methods for HCI*. New York: Springer.

Suggested Readings

1. Rodrigues, J., Cardoso, P., Monteiro, J., & Figueiredo, M. (2016). *Handbook of research on human-computer interfaces, developments, and applications*. Hershey: IGI Global.
2. Picard, R. W. (2000). *Affective computing*. Cambridge: The MIT press.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: strategies for effective human-computer interaction*. New York: Pearson.
4. Dix, A., Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. New York: Pearson Education

CSEC-7403 Advanced Web Development 3 (3+0)

To introduce students with Web Systems and Technologies Implement HTML5 coding solutions using HTML5 elements, attributes and values. Apply CSS33 functionality to Web documents using various properties, selectors and techniques. Integrate basic JavaScript coding into a Web page to create HTML5 APIs. Consider HTML5 and CSS33 techniques to apply to both traditional and mobile delivery platforms, including mobile apps. Transform traditional Web pages into mobile Web pages. Understanding PHP frameworks and advanced technologies. Students will learn the fundamentals of JavaScript code, and then get into jQuery. jQuery is an industry standard framework that lets students quickly and easily write powerful JavaScript. Students will learn how to use some popular jQuery plugins, and gain an understanding of how plugins work, so you can use any plugin. Something new in JQuery. Mobile websites and mobile apps development. Upon successful completion of course students will be able to apply learnt PHP, CSS33, boot strapping and WSDL along with advanced JQuery concepts.

Contents

1. Advanced HTML5 concepts: Structure Elements, Validating HTML5 Code, The <video> Element, The <audio> Element, Introduction to HTML5 Forms, Cross-Browser Compatible HTML5 Forms, New Form Elements in HTML5, HTML5 Global Attributes for Form Elements.
2. HTML5 APIs, Document Object Model (DOM), The Canvas API, The Offline AppCache API, The Geolocation API, The Drag-and–Drop API, The File API, The History API, Retrieving Data with *XMLHttpRequest.*
3. Advanced CSS33 concepts: Cascading Style Sheets (CSS3), Style Guides, CSS3 and HTML, CSS3 Terms and Syntax, Applying CSS3 Styles, Page Layout with CSS3, CSS3 Positioning Schemes, The CSS3 Box Model.
4. Introduction to CSS33, CSS33 Selectors and Properties, CSS33 Background Properties, CSS33 Border Properties, CSS33 Font Properties, CSS33 Text Effects,
5. Introduction to Advanced CSS33 Techniques, CSS33 2D and 3D Transformations, CSS33 Transitions, CSS33 Animations, CSS33 User Interfaces, Creating Menus and Buttons with CSS33.
6. Advanced JavaScript concepts: Learning JavaScript Design Patterns.
7. Advanced PHP concepts: functions, date and time, debugging and logging, string manipulation, database connectivity methods
8. PHP extensions and application repository –PEAR, Smarty Templates, code reuse, feeding a function, return data from a function, Runtime error handling using PHP exception handling mechanism
9. Web services including SOAP, NU SOAP and WSDL, popular frameworks including CakePHP, Yii, Zend and Codelgniter. Advanced JQuery concepts: Bind/Unbind, Live/Die, Ajax Queue/Sync
10. Migration to Mobile Devices, Mobile Web Sites vs. Mobile Apps, Designing Web Sites for Mobile Devices, Page Layout for Mobile Devices, Navigation and Hyperlinks for Mobile Devices

Pre-Requisites: Web Systems and Technologies.

Recommended Texts

1. Castro, E., & Hyslop, B. (2015). *HTML5 a CSS3*. Prague: Albatros Media.
2. Negrino, T, & Smith, D. (2013). *Dreamweaver CC: visual quickstart guide*. San Francisco: Berkeley Peachpit Press.

Suggested Readings

1. Pollock, J. (2001). *JavaScript: a beginner's guide*. New York: McGraw-Hill.

CSEC-7404 **Advanced Web Engineering** **3 (3+0)**

Web Engineering introduces a structured methodology utilized in software engineering to Web development projects. The course addresses the concepts, methods, technologies, and techniques of developing Web sites that collect, organize and expose information resources. Advanced Web Engineering can provide you with an unprecedented level of service and expertise for all of your marketing campaigns like email and social media marketing, event management, along with the ability to create offers and local deals for your business. It is a study of the concepts, principles, techniques and methods of Web engineering. The course is aimed to provide students with conceptual understanding required to develop web applications and web services according to international standards**.** It is an extended version of Web Engineering with some material covered in greater depth. By the end of this course students will be able to analyze and design comprehensive systems for the creation, dissemination, storage, retrieval, and use of electronic records and documents.

Contents

1. The Need for Web Engineering
2. Web Effort Estimation
3. Web Productivity Measurement and Benchmarking
4. Web Quality
5. Web Usability
6. Web System Reliability and Performance
7. Web Application Testing
8. An Overview of Process Improvement in Small Settings: Initiating SPI Efforts, Process Improvement Cycle, Process Assessments, Implementation in Small Settings.
9. Conceptual Modelling of Web Applications: The OOWS Approach: Introduction, A Method to Model Web Applications, A Strategy to Develop the Web Solution, Case Study: *Valencia CF Web Application.*
10. Model-Based Web Application Development: The OOHDM approach, Building an Online CD Store with OOHDM, From Design to Implementation, Discussion and Lessons Learned.
11. W2000: A Modeling Notation for Complex Web Applications: Introduction, Modelling Elements, Models, Example Application: Information Model, Navigation Model, Presentation Model, 4 Service Model.
12. Statistics Analysis
13. W3C Web Content Accessibility Guidelines
14. Internationalization

.

Recommended Texts

1. Roger Pressman and David Lowe. (2008). Web engineering: A practitioner's approach, (1st ed.). New York: McGraw-Hill.

Suggested Readings

1. Emilia Mendes and Nile Mosley. (2010). *Web engineering*, (1st ed.). New York: Springer.

CSEC-7405 **Agent Based Modeling** **3 (3+0)**

Agent Based Models are computer models that attempt to capture the behavior of individuals within an environment. We define traits and initial behavior rules of an agent that organize their actions and interactions. Stochasticity plays an important part in determining which agents interact and how agents make decisions. Students will be able to learn agent-based modelling and social network theory along with biological systems. Because agents carry knowledge with them as they move, they can utilize that knowledge and history as they travel around experiencing new areas and meeting other agents. Students will learn how to build a model from the ground up and how to analyze and understand the results of a model using the NetLogo programming language. We will also discuss how to build models that are sound and rigorous, a self-forming neighborhood model, cellular automata, and current research in this area. Course will also include few practical exercises.

Contents

1. Introduction to agent-based modelling.
2. Introduction to Net Logo.
3. Complexity in Social Worlds.
4. Net Logo Commands.
5. Net Logo Procedures.
6. Model properties.
7. Why agent-based objects?
8. Agents, environments, and timescales.
9. Biological systems: fireflies
10. Flocking, slime mold, bees, ants (flocking behaviour slime mold).
11. Biological systems: predator/prey
12. Debugging (Verification and validation).
13. Social systems: segregation, Schelling,
14. Micro motives and Macro behaviour.
15. A self-forming neighbourhood model.
16. Cellular automata.
17. Critical phenomena. Sand piles.
18. Current research topics in Agent Based Modelling.

.

Recommended Texts

1. Nigel Gilbert. (2008). *Agent-based models*, Thousand Oaks: SAGE Publications.

CSEC-7406 Big Data Analytics 3 (3+0)

The main aim of the course is to frame big data jobs as Apache Spark Scripts. Students should be capable enough to optimize an algorithm by partitioning it across worker nodes. At the end of this course, student will become familiar with the fundamental concepts of Big Data management and analytics. Also, they will become competent in developing distributed code using Scala language and Apache Spark API. The course will provide enough concepts to deploy a spark cluster and run jobs on the cluster. Course will also cover GraphX, MLib, SparkSQL libraries to study their application in social networks and link prediction. This course is for those new to data science. No prior programming experience is needed, although the ability to install applications and utilize a virtual machine is necessary to complete the hands-on assignments. This course focuses on RDD based algorithm design, with an emphasis on text processing algorithms common in natural language processing, information retrieval, and machine learning.

Contents

1 Big Data: Issues and challenges
2 Big Data Tools: An overview
3 Functional Programming Paradigm
4 Scala: Basics, Functions, and Data Structures
5 Scala Practical Exercise
6 Apache Spark & RDD
7 Deploy Apache Spark Cluster
8 Implement and Run Apache Spark Job on Cluster
9 Broadcast Variables and Accumulative Variables in Apache Spark
10 Island, Global, and Grid Models
11 Translating Embeddings for Modeling Multi-relational Data
12 Scalable Genetic Algorithms using Apache Spark
13 ACO for Link Prediction
14 Using Neighborhood for Link Prediction Problems
15 GraphX
16 MLib
17 SparkSQL

Recommended Texts

1. Kenneth Cukier and Viktor Mayer-Schönberger (2014). *Big data: a revolution that will transform how we live, work, and think*. London: Eamon Dolan/Mariner Books
2. Jimmy Lin and Chris Diyer (2010). *Data-intensive text processing with mapreduce*. California: Morgan & Claypool Publishers

Suggested Readings

1. Andy Konwinski, Holden Karau, Matei Zaharia, and Patrick Wendell (2015). *Learning spark: lightning-fast big data analysis*. Sebastopol: O'Reilly Media.
2. Anand Rajaraman and Jeff Ullman (2011), *Mining of massive dataset*s. Cambridge: Cambridge Press.

CSEC-7407 **Bioinformatics** **3 (3+0)**

This course aims to introduce students to the current bioinformatics concepts and their implementations. A student completing a major in Bioinformatics shall be able to apply knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics, existing software effectively to extract information from large databases and to use this information in computer modeling, problem-solving skills, including the ability to develop new algorithms and analysis methods, an understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory. Scholars are able to learn and understand Distance-Based Methods of Phylogenetics, Phylogenetic Trees, Terminology of Tree Reconstruction, Rooted and Unrooted Trees, Gene vs. Species Trees, Character and Distance Data. Distance Matrix Methods UPGMA. Estimation of Branch Lengths, Transformed Distance Method, Neighbor's Relation Method, Neighbor-Joining Methods. Maximum Likelihood Approaches. Multiple Sequence Alignments. Term papers are part of course.

Contents

1. Molecular Biology and Biological Chemistry: The Genetic Material: Nucleotides, Orientation, Base Pairing, Central Dogma of Molecular Biology. Gene Structure and Information Content: Promoter Sequences, The Genetic Code. Introns and Exons. Protein Structure and Function: Primary, Secondary, Tertiary and Quaternary Structure. Nature of Chemical Bonds: Anatomy of Atom, Valence. Electronegativity, Hydrophilicity and Hydrophobicity. Molecular Biology Tool.
2. Data Searches and Pairwise Alignments: Dot Plots. Simple Alignments. Scoring. Gaps: Simple Gap Penalties, Origination and Length Penalties. Scoring Matrices. Dynamic Programming: The Needleman AndWunsch Algorithm. Local and Global Alignments: Global and Semi-Global Alignments, The Smith-Waterman Algorithm. Database Searches: BLAST and Its Relatives, Other Algorithms.
3. Substitution Patterns: Patterns of Substitutions Within Genes: Mutation Rates. Functional Constraint. Synonymous Vs. Non-synonymous Changes, Indels and Psuedogenes, Substitutions Vs. Mutations, Fixation. Estimating Substitution Numbers: Jukes/Cantor Model, Transitions and Transversions. Kimura's Two-Parameter Model, Models with Even More Parameters, Substitutions between Protein Sequences. Variations in Substitution Rates between Genes. Molecular Clocks.
4. Character-Based Approaches to Phylogenetics: Parsimony: Informative and Uninformative Sites, Unweighted Parsimony, Weighted Parsimony. Inferred Ancestral Sequences. Searching Strategies.
5. Genomics and Gene Recognition: Prokaryotic Genomes. Prokaryotic Gene Structure: Promoter Elements, Serial Analysis of Gene Expression, Microarrays. Transposition. Repetitive Elements.
6. Protein Folding: Polypeptide Composition: Amino Acids. Backbone Flexibility, Phi and Psi. Secondary Structure: Accuracy of Predictions, Chou-Fasman/GOR Method. Tertiary and Quaternary Structure: Hydrophobicity, Disulfide Bonds,
7. Proteomics: From Genomes to Proteomes. Protein Classification: Enzyme Nomenclature, Families and Super-Families, Folds. Experimental Techniques.

Recommended Texts

1. Krane, D. E. (2002). *Fundamental concepts of bioinformatics*. New York: Pearson Education.

2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

ITEC-7401 Business Process Management Trends & Technologies 3 (3+0)

Business process and business process management concepts have matured over the years and new technology, concepts, standards, and solutions appear. The inclusion of big data and analytics have brought about enormous changes in the way's organizations operate and provide training to their employees. The deployment of BPM software generates a massive amount of data, which can be used for process optimization and performance improvement. BPM analytics is widely used for decision making, optimization of internal operations, and monitoring the performance of the process. The course will cover topics fundamentals and principles of Business Process Management. The course is designed to achieve following objectives: Understand the key terms and concepts in BPM, Learn the major methodologies and techniques for implementing BPM, Discover the various technologies that support BPM, Learn what a BPM management and process-centric organization is and how it, works, Understand the metrics and measurements critical to managing processes, Learn how to identify critical processes

Contents

1. Introduction: Motivation and Definitions, Business Process Lifecycle, Classification of Business Processes, Goals, Structure, and Organization.
2. Evolution of Enterprise Systems Architectures: Traditional Application Development, Enterprise Applications and their Integration, Enterprise Modeling and Process Orientation, Workflow Management, Enterprise Services Computing.
3. Business Process Modeling: Foundation, Conceptual Model and Terminology, Abstraction Concepts, From Business Functions to Business Processes, Activity Models and Activity Instances, Process Models and Process Instances.
4. Process Orchestrations: Control Flow Patterns, Petri Nets, Event-driven Process Chains, Workflow Nets, Graph-Based Workflow Language, Business Process Model and Notation.
5. Process Choreographies: Motivation and Terminology, Development Phases, Process Choreography Design, Process Choreography Implementation, Service Interaction Patterns, Choreography Modeling in BPMN.
6. Properties of Business Processes: Data Dependencies, Object Lifecycle Conformance, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness, Soundness Criteria Overview.
7. Business Process Management Architectures: Workflow Management Architectures, Flexible Workflow Management.

Recommended Texts

1. Mathias Weske. (2012). *Business process management: concepts, languages, architectures*, (2nd ed.). New York: Springer.
2. Yvonne Lederer Antonucci. (2009). *Business process management common body of knowledge*. Scotts Valley: CreateSpace Independent Publishing Platform.

Suggested Readings

1. Paul Harmon. (2007). *Business process change: a guide for business managers and BPM and six sigma professionals* (2nd ed.). Burlington: Morgan Kaufmann Publishers.
2. Jörg Becker, Martin Kugeler and Michael Rosemann, Process. (2011). *Management: a guide for the design of business processes*, (2nd ed.). New York: Springer.

CSEC-7408 Cloud Computing 3 (3+0)

Cloud Computing has transformed the IT industry by opening the possibility for infinite or at least highly elastic scalability in the delivery of enterprise applications and software as a service (SaaS). Amazon Elastic Cloud, Microsoft's Azure, Google App Engine, and many other Cloud offerings give mature software vendors and new start-ups the option to deploy their applications to systems of infinite computational power with practically no initial capital investment and with modest operating costs proportional to the actual use. This course gives an introduction to cloud computing and related techniques, issues, ecosystem and case studies. Students will learn and understand about such fundamental distributed computing "concepts" for cloud computing, how these techniques work inside today's most widely-used cloud computing systems and various research papers will be studied. Students will learn Datacenter Architectures, Cloud Stack , Technology Trends, Consistency, Availability, Partitions, Cluster File Systems, Data-flow Computation Frameworks, Key-Value Store and Interactive Query Systems, Big Data in the Clouds, Geographic distributed Storage, Programming Languages for the Cloud, DBases in the Cloud, In-Memory Frameworks, Google file system, Hadoop file system, MapReduce, OSes and Clouds Networking: topologies, Networking: Traffic Management, Networking: Transport Protocol Improvements, Security, Scheduling and Resource Management in clouds, Software Level Agreements.

Contents

1. Overview of Distributed Computing, Emergence of Cloud Computing, Global Nature of the Cloud, Cloud-Based Service Offerings, Grid Computing, Reliability of Cloud Model, Benefits of Cloud Model, Legal Issues,
2. Key Characteristics of Cloud Computing, Challenges for the Cloud. The Evolution of Cloud Computing.
3. Web Services Delivered from the Cloud: Communication-as-a-Service (CaaS), Infrastructure-as-a-Service, Monitoring-as-a-Service (MaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS),
4. Storage: HDFS, NAAS, SAN, Distributed Graph Processing, MapReduce, MapReduce Paradigm, MapReduce Examples, Emerging Paradigms, Distributed Graph Processing, Hadoop Scheduling, Dominant-Resource Fair Scheduling, Building Cloud Networks. Virtualization. Federation, Presence, Identity, and Privacy in the Cloud. Security in the
5. Cloud. Common Standards in Cloud Computing. End-User Access to Cloud Computing. Mobile Internet Devices and the Cloud.

Pre-Requisites: Computer Networks

Recommended Texts

1. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. Boca Raton: CRC press.
2. Rothon, J. (2009). *Cloud computing explained: Implementation handbook for enterprises*.

Suggested Readings

1. Linthicum, D. S. (2009). *Cloud computing and SOA convergence in your enterprise: a step-by-step guide*. New York: Pearson Education., Boston: Addison-Wesley Professional.
2. Sosinsky, B. (2010). *Cloud computing bible* (vol. 762). Hoboken: John Wiley & Sons.

CSEC-7410 **Computer Vision** **3 (3+0)**

Computer Vision is a field of Artificial Intelligence and Computer Science that aims at giving computers a visual understanding of the world. Computer vision spans all tasks performed by biological vision systems, including "seeing" or sensing a visual stimulus, understanding what is being seen, and extracting complex information into a form that can be used in other processes. The goal of Computer Vision is to emulate human vision using digital images through main processing components. The course concentrates on the knowledge of Computer Vision and its importance in Computing area having multiple applications. By the end of the course students will be able to develop basic methods for applications that include finding known models in images, depth recovery from stereo, camera calibration, three-dimensional image reconstruction, rendering based on images, Laws' Texture Energy Measure image stabilization, Syntactic Texture Description Methods, automated alignment, tracking, boundary detection, and recognition. Course will also include few practical exercises.

Contents

1. Computer Vision an Introduction.
2. Image formation.
3. Image Processing.
4. Feature Detection and Matching.
5. Feature-based Alignment.
6. Image Stitching.
7. Dense Motion Estimation.
8. Structure from Motion.
9. Recognition.
10. Computational Photography.
11. Stereo Correspondence.
12. 3D Reconstruction.
13. Image-based Rendering.
14. Statistical Texture Description, Methods Based on Spatial Frequencies, Co-occurrence Matrices, Edge Frequency, Primitive Length (Run Length).
15. Laws' Texture Energy Measure
16. Fractal Texture Description, Multiscale Texture Description – Wavelet Domain Approaches, other Statistical Methods of Texture Description
17. Syntactic Texture Description Methods, Shape Chain Grammars, Graph Grammars, Primitive Grouping in Hierarchical Textures, Hybrid Texture Description methods, Texture Recognition Method Applications.

Pre-Requisites: Linear Algebra

Recommended Texts

1. Richard Szeliski, (2011). *Computer vision algorithms and applications*, New York: Springer.
2. Milan Sonka. (2008). *Image processing, analysis, and machine vision*, (3rd ed.). Washington: CL Engineering

Suggested Readings

1. Dr Simon J. D. Prince. (2012). *Computer vision: models, learning, and inference*, Cambridge: Cambridge University Press.

ITEC-7411 **Digital Image Processing** **3 (3+0)**

This course provides an introduction to basic concepts, methodologies and algorithms of digital image processing including image sampling and quantization, color, point operations, segmentation, morphological image processing, linear image filtering and correlation, image transforms, eigen images, multiresolution image processing, noise reduction and restoration, feature extraction and recognition tasks, image registration. Emphasis is on the general principles of image processing. Students learn to apply material by implementing and investigating image processing algorithms in Matlab. Scholars are able to learn different image processing techniques and compare them. Different recent state of the art articles are given for term paper and the end of semester a term paper is submitted on the following topics Spatial Frequency, Fourier Theory, the Discrete Fourier Transform, Investigating Spectra Filtering of Images, Deconvolution , Storage Media, File Formats, The PBM, PGM, PPM Formats, The Portable Network Graphics (PNG) Format in a group format. Presentation of term paper is also part of course evaluation.

Contents

1. What Is Digital Image Processing?
2. The Origins of Digital Image Processing,
3. Examples of Fields that Use Digital Image Processing.
4. Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition.
5. Spatial Frequency, Fourier Theory, the Discrete Fourier Transform, Investigating Spectra.
6. Filtering of Images, Deconvolution.
7. Color Fundamentals, Color Models.
8. Storage Media, File Formats, The PBM, PGM, PPM Formats, The Portable Network Graphics (PNG) Format.
9. Segmentation Fundamentals, Point Line and Edge Detection.
10. Morphological Image Processing Introduction, Basic Concepts, Fundamental Operations.
11. Compound Operations.
12. Morphological Filtering, Greyscale Morphology.
13. Image Compression Introduction, Redundancy, Performance Characterisation.
14. Lossless Compression Techniques
15. Lossy Compression Techniques, Compression of Moving Images.

.*Recommended Texts*

1. Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*. New York: Pearson Education.
2. Efford, N. (2000). *Digital image processing: a practical introduction using java* (with CD-ROM). Boston: Addison-Wesley Longman Publishing Co.

Suggested Readings

1. Ross, L., & Russ, J. C. (2011). *The image processing handbook. Microscopy and Microanalysis*.

Cambridge: Cambridge University Press.

ITEC-7402 Distributed Database Systems 3 (3+0)

A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sited that don't share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database. The increased capabilities of a collection of logically interrelated databases distributed over a computer network enable scalable data processing. This course addresses the components of these systems, covering the main topics The aim of the course is to enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficiencies of the centralized database systems; introduce basic principles and implementation techniques of distributed database systems, expose active research issues in distributed database systems.

Contents

1. Distributed Data Processing, Distributed Database Systems, Data Delivery Alternatives, Promises of DDBSs, Complications Introduced by Distribution, Design Issues, Distributed DBMS Architecture.
2. Distributed Database Design: Top-Down Design Process, Fragmentation, Allocation, Data Directory.
3. Database Integration: Bottom-Up Design Methodology, Schema Matching, Schema Integration, Schema Mapping, Data Cleaning.
4. Data and Access Control: View Management, Data Security, Semantic Integrity Control.
5. Overview of Query Processing: Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing.
6. Query Decomposition and Data Localization: Query Decomposition, Localization of Distributed Data.
7. Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.
8. Multidatabase Query Processing: Issues in Multidatabase Query Processing, Multidatabase Query Processing Architecture, Query Rewriting Using Views, Query Optimization and Execution, Query Translation and Execution.
9. Transaction Management: Definition of a Transaction, Properties of Transactions, Types.

Pre-Requisites: DBMS

Recommended Texts

1. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (Vol. 2). New York: Prentice Hall.
2. Rahimi, S. K., & Haug, F. S. (2010). *Distributed database management systems*: *a practical approach*. Hoboken: John Wiley & Sons.

Suggested Readings

1. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. New York: Prentice-Hall.

CSEC-7413 **Distributed Systems** **3 (3+0)**

The aim of the course is to enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficiencies of the centralized database systems; introduce basic principles and implementation techniques of distributed database systems, expose active research issues in distributed database systems. Advance topics on distributed systems like Distributed Concurrency Control: Serializability Theory, Taxonomy of Concurrency Control Mechanisms, Locking-Based Concurrency Control Algorithms, Timestamp-Based Concurrency Control Algorithms, Optimistic Concurrency Control Algorithms, "Relaxed" Concurrency Control, Distributed DBMS Reliability: Reliability Concepts and Measures, Failures in Distributed DBMS, Local Reliability Protocols, Distributed Reliability Protocols, Dealing with Site Failures, Network Partitioning, Architectural Considerations are also part of this course. Term system is assigned to each student in this course based on the relevant topics that includes Distributed Data Processing, Complexity of Relational Algebra Operations, Optimization of Distributed Queries, Multidatabase Query Processing and Data Replication: Consistency of Replicated Databases

Contents

1. Distributed Data Processing, Distributed Database Systems, Data Delivery Alternatives, Promises of DDBSs, Complications Introduced by Distribution, Design Issues, Distributed DBMS Architecture.
2. Distributed Database Design: Top-Down Design Process, Fragmentation, Allocation, Data Directory.
3. Database Integration: Bottom-Up Design Methodology, Schema Matching, Schema Integration, Schema Mapping, Data Cleaning.
4. Data and Access Control: View Management, Data Security, Semantic Integrity Control.
5. Overview of Query Processing: Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing.
6. Query Decomposition and Data Localization: Query Decomposition, Localization of Distributed Data.
7. Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.
8. Multidatabase Query Processing: Issues in Multidatabase Query Processing, Multidatabase Query Processing Architecture, Query Rewriting Using Views, Query Optimization and Execution, Query Translation and Execution.
9. Transaction Management: Definition of a Transaction, Properties of Transactions, Types.

Pre-Requisites: DBMS

Recommended Texts

1. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.
2. Rahimi, S. K., & Haug, F. S. (2010). *Distributed database management systems*: *a practical approach*. Hoboken: John Wiley & Sons.

Suggested Readings

1. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. New York: Prentice Hall.

ITEC-7403 Enterprise Data Center Design and Methodology 3 (3+0)

As data centers are inevitably growing more complex and larger, it brings many challenges to the deployment, resource management and service dependability. As data center becomes more and more central in the present age of internet communication, both research and operations communities have begun to explore how to better design and manage them. There are some materials providing guideline for data center design. This course provides a scalable, modular methodology for designing data centers of any size and capability Includes design techniques for accurate planning based on data center capacities It covers all aspects of data center design from site selection to network connectivity. The fundamental design principles take a simple, flexible, and modular approach based on accurate, real-world requirements and capacities. This approach contradicts the conventional method of using square footage to determine basic capacities like power and cooling requirements. The course is aimed to introduce students with the essential knowledge about Data Center and methodologies that optimizes availability, scalability, and performance for that environment and provides centralized management services.

Contents

1. Overview of Data Centers
2. Introduction to Data Centers,
3. Application Architecture Models,
4. Data Center Architecture,
5. Data Center Services.
6. Data Center Requirements
7. Data Center Pre-requisites
8. Budget Constraints, Selecting Geographical Location, Retrofitting.
9. Server Architecture
10. Application Architectures
11. Network Infrastructure.
12. Data Center Security
13. Designing the Data Center Infrastructure
14. Integrating Security into the Infrastructure
15. Server Capacity Planning
16. Best Practices in IT

.*Recommended texts*

1. Mauricio Arregoces and Maurizio Portolani. (2003). *Data center fundamentals*, Indianapolis: Cisco Press.
2. UrsHoelzleand Luiz Andre Barroso. (2009). *The datacenter as a computer: An introduction to the design of warehouse-scale machines*, (1st ed.). San Rafael: Morgan and Claypool Publishers.

Suggested Readings

1. Kailash Jayaswal. (2005). *Administering data centers: servers, storage, and voice over IP*, (1st ed.). Hoboken: Wiley.

CSEC-7414 **Expert Systems** **3 (3+0)**

This is an especially powerful approach for problems that change often or where solutions involve application of human knowledge, rather than intricate calculations. Today, they are used in business, science, engineering, manufacturing, etc. Example applications include: computer configuration, fault diagnosis, computer-aided instruction, data interpretation, planning and prediction, and process control. The course concentrates on an analysis of the architecture, knowledge and problem-solving style of expert system. Students will learn distinguishing features of Expert Systems, theoretical features, basic form of inference, basic components of expert systems, knowledge engineering tools. Students will be able to explain and describe the concepts central to the creation of knowledge bases and expert systems, to apply the tools and the processes used for the creation of an expert system. Student will know methods used to evaluate the performance of an expert system, conduct an in-depth examination of an existing expert system with an emphasis on basic methods of creating a knowledge base. Students will be able to examine properties of existing systems in a case-study manner, comparing differing approaches.

Contents

1. Introduction to Expert Systems.
2. Major Characteristics of Expert Systems.
3. Expert System Architecture, Knowledge Representation.
4. Prolog Programming, Developing Meta Interpreters in Prolog.
5. Inference Techniques, MYCIN.
6. Rule-Based Expert Systems.
7. Backward-Chaining Rule-Based Systems.
8. Designing Backward-Chaining Rule-Based Systems.
9. Forward-Chaining Rule-Based Systems.
10. Designing Forward-Chaining Rule-Based Systems.
11. Frame-Based Expert Systems.
12. Designing a Frame-Based Expert Systems.

Recommended Texts

1. Durkin, J. (1998). *Introduction to Expert Systems*, (3rd ed.). Boston: Addison-Wesley.
2. Ken Pedersen. (1989). *Expert Systems Programming: Practical Techniques for Rule-Based Systems*, (1st ed.). Hoboken: Wiley.

Suggested Readings

1. Patterson, D. W. (1990). *Introduction to artificial intelligence and expert systems*. New York: Prentice Hall.
2. Nikolopoulos, C. (1997). *Expert systems: introduction to first and second generation and hybrid knowledge-based systems*. New York: Marcel Dekker, Inc.
3. Rich, E., & Knight, K. (1992). *Artificial Intelligence: Instructor's Manual*. New York: McGraw-Hill.
4. Jean-Louis Lauriere. (1990). *Problem Solving and Artificial Intelligence*. New York: Prentice Hall.

CSEC-7415 **Formal Methods** **3 (3+0)**

Modern software development inevitably requires the design and analysis of a number of different artifacts. Formal methods allow the mathematically precise formulation of some of these artifacts. The purpose of this course is to learn how to specify behavior of systems and to experience the design of a system where you can prove that the behavior is correct. This course is an introduction to the use of formal methods for the specification, de-sign, and automatic analysis of software systems. Focus is on theory for specification, validation and verification of network and network-based service functionality specified by communicating state machines, protocol algebraic formulas, and temporal logic descriptions. The theory covers verification by process algebra, temporal logic, rewriting logic and reasoning on UML constraints. With respect to verification tools, model checkers, theorem provers, SAT and SMT solvers will be discussed. More specifically, it deals with formal vs informal requirements, state transition diagrams, predicates, various elements of Z notational language, schema and schema calculus, formal reasoning with famous 8-queen problem, formal intuitions, and other state-of-the-art research in this area.

Contents

1. Formal methods: What is FM? What FM are not, When, How, and Why use FM?
2. Popular Fallacies and Alternatives.
3. Formal Methods and Project Management: Gathering Requirement, From Information Requirement to Formal Specifications,
4. Validating Formal Specifications.
5. Introducing Z: What is Z? Informal Requirements, Data Flow Diagrams, State Transition Diagram.
6. State Transition Diagram, State Transition tables. Introducing schemas: Basic Types and Abbreviations, Axiomatic Descriptions, State Schemas, Operation Schemas, Implicit preconditions, Schema Calculus.
7. Elements of Z: Sets and Types, Declarations, Variables, Expressions, Operators, Predicates, Equations and Laws. Structure: Tuple, Records, Relations, Tables, Databases, Pairs and Binary Relations, Functions, Sequences, Operators.
8. Logic: Basic Predicates, Using Predicates in Z, Relations as Predicates, Logical Connectives, Logic and Natural Language, Quantifiers, Z and Boolean Types, Predicates and undefined Expressions. Synthesis: Set Comprehensives, Lambda Expressions, Formal Specifications
9. Schemas and schema calculus: Conjunctions and Disjunctions, Other Schema Operators. Schema types and Bindings: Generics & Free Types.
10. Formal Reasoning: Calculation and proof, Laws, Checking Specifications, Preconditions
11. Graphical User Interface, Converting Z Specification into Code.

Recommended Texts

1. Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., & Schnoebelen, P. (2013). *Systems and software verification: model-checking techniques and tools*. New York: Springer Science & Business Media.
2. Jacky, J. (1997). *The way of Z: practical programming with formal methods*. Cambridge: Cambridge University Press.

Suggested Readings

1. Diller, A. (1994). *Z: an introduction to formal methods*. Hoboken: John Wiley & Sons, Inc.
2. Clarke Jr, E. M., Grumberg, O., Kroening, D., Peled, D., & Veith, H. (2018). *Model checking*.

Cambridge: The MIT press.

CSEC-7416 Functional Programming 3 (3+0)

Broadly speaking, functional programming is a style of programming in which the primary method of computation is the application of functions to arguments. Among other features, functional languages offer a compact notation for writing programs, powerful abstraction methods for structuring programs, and a simple mathematical basis that supports reasoning about programs. The students will gain an applied understanding of the concepts and practice of functional programming. The goal of the course is to delve deeper in to the principles of program design, implementation and understanding. The course aims to use the language to implement algorithms and data types to solve problems; to explain common errors in sample programs and offer corrections to discuss functional programming and functional programs. On completion of the course, the student should be able to list and define the fundamental concepts of functional programming, manually execute a given (simple) functional program, manually infer the type of a given (simple) functional program, implement (simple) algorithms and data structures as functional programs and design (large) functional programs that are modular etc.

Contents

1. Introducing functional programming: Computers and modeling, Pictures and functions, Types, The Haskell programming language, Haskell Interpreter, Implementing a Prime Number Test,
2. Haskell Type Declarations, Identifiers in Haskell, Haskell Types, The Prime Factorization Algorithm, The map and filter Functions, Haskell Equations and Equational Reasoning.
3. Mathematical Objects: Logical Connectives and their Meanings
4. Lambda Abstraction, Definitions and Implementations, Abstract Formulas and Concrete Structures, Logical Handling of the Quantifiers, Quantifiers as Procedures.
5. The Use of Logical Proof: Proof Style, Proof Recipes, Rules for the Connectives, Rules for the Quantifiers, Strategic Guidelines, Reasoning and Computation with Primes.
6. Sets, Types and Lists: Sets, Paradoxes, Types and Type Classes, Special Sets, Algebra of Sets
7. Comprehension and Database Query, Using Lists to Represent Sets, A Data Type for Sets.
8. Relations: The Notion of a Relation, Properties of Relations, Implementing Relations as Sets of Pairs.
9. Implementing Relations as Characteristic Functions, Equivalence Relations
10. Functions: Basic Notions, Surjections, Injections, Bijections, Function Composition, Inverse Function, Partial Functions, Functions as Partitions, Products, Congruences.
11. Induction and Recursion: Mathematical Induction, Recursion over the Natural Numbers
12. Induction and Recursion over Trees, Induction and Recursion over Lists, Some Variations on the Tower of Hanoi, Induction and Recursion over Other Data Structures.
13. Polynomials: Difference Analysis of Polynomial Sequences, Gaussian Elimination, Polynomials and the Binomial Theorem, Polynomials for Combinatorial Reasoning.
14. Corecursion: Corecursive Definitions, Processes and Labeled Transition Systems
15. Power Series and Generating Functions, Exponential Generating Functions.

Recommended Tests

1. Sankel, D. (2006). *The haskell road to logic, maths and programming*. London: King's College Publications.

Suggested Readings

1. MacLennan, B. J. (1990). *Functional programming: practice and theory*. Boston: Addison-Wesley Longman Publishing Co., Inc

CSEC-7417 **Fuzzy Logic** **3 (3+0)**

This course will start with a brief introduction to fuzzy sets. The differences between fuzzy sets and crisp sets will be identified. Various terms used in the fuzzy sets and the grammar of fuzzy sets will be discussed, in detail, with the help of some numerical examples. It is an important area having multiple applications in Computing. The course will introduce students with key concepts of Fuzzy Logic. Students will learn new paradigm fuzzy access to some part of the classical mathematics and logic. To show how to model fuzzy sets, how to handle with arithmetic of fuzzy quantities, and to acquire operations with fuzzy relations. To understand the mechanism of fuzzy reasoning and the role the essential models in fuzzy inference. Upon successful completion of course students will be able to understand fuzzy set theory, recognize fuzzy logic membership function, recognize fuzzy logic fuzzy inference systems, make applications on Fuzzy logic membership function and fuzzy inference systems, use the fuzzy set theory on the statistical method which is given.

Contents

1. Fuzziness: Introduction, Examples of Fuzziness, Modeling of Fuzziness.
2. Operations on Fuzzy Sets, Fuzziness as Uncertainty
3. Algebra of Fuzzy Sets: Boolean Algebra and Lattices.
4. Equivalence Relations and Partitions, Composing Mappings,
5. Isomorphism, and Homomorphisms
6. Alpha Cuts, Images of Alpha Level Sets.
7. Fuzzy Quantities, Fuzzy Numbers, Fuzzy Intervals.
8. Basic Connectives: t – Norms, Generators of t – Norms.
9. Isomorphisms of t – Norms, Negations, t – Conorms, Strict De – Morgan Systems, Nilpotent De Morgan Systems, Non-uniqueness of Negations in Strict De Morgan Systems.
10. Fuzzy Implications, Averaging Operators and Negations
11. De Morgan Systems with Averaging Operators.
12. Power of t-Norms, Sensitivity of Connectives.
13. Fuzzy Relations: Binary Fuzzy Relations, Operations on Fuzzy Relations, Approximate Reasoning, Approximate Reasoning in Expert Systems, Modulus Ponens.
14. Universal Approximations: Fuzzy Rule Bases, Desing Methodologies, Approximation Capabilities
15. Partial Knowledge: Belief Functions, Indicence Algebras,
16. Monotonicity, Beliefs, Densities and Allocations, Möbius Transform,
17. Reasoning with Belief Functions, Decision Making with Belief Functions.

Recommended Texts

1. Walker, E., & Nguyen, H. T. (2006). *A first course in fuzzy logic.* Boca Raton: CRC press.

Suggested Readings

1. Zimmermann, H. J. (2011). *Fuzzy set theory—and its applications*. New York: Springer Science & Business Media.
2. Dubois, D., & Prade, H. (Eds.). (2012). *Fundamentals of fuzzy sets* (vol. 7). New York: Springer Science & Business Media.
3. Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (vol. 4). Hoboken: Prentice hall.

CSEC-7418 **Game Theory** **3 (3+0)**

This course provides an introduction to Game Theory. Game Theory is a mathematical framework which makes possible the analysis of the decision-making process of interdependent subjects. It is aimed at explaining and predicting how individuals behave in a specific strategic situation, and therefore help improve decision making. Students will learn how to represent an economic situation as a game and how to analyze it using different equilibrium concepts proposed in the literature, the prominent one being the Nash equilibrium. The course concentrates on strategic interaction under incomplete information and modify the Nash equilibrium concept to include the uncertainty of the players about some of the parameters of the game. Often, an equilibrium concept fails to provide a unique solution to the game. At the end students will deal with the problem of indeterminacy in games in extensive form and introduce refinements of the Nash equilibrium. Course will also include few practical exercises.

Contents

1. Introduction to Game Theory, The Two-Person, Zero-Sum Game with Equilibrium Points, The General, Two-Person, Zero-Sum Game.
2. Utility Theory, The Two Person, Non-Zero-Sum Game.
3. The N-Person Game
4. Strategic Games: Basic Concepts, Iterated Elimination of Strategies, Mixed Extension.
5. Variations on the Definition of Strategic Games, Mechanism Design.
6. Pre-Bayesian Games
7. Infinite Games and Automata Theory: Basic Notations and Definitions, Transformation of Winning Conditions, Tree Automata, Beyond Finite Automata.
8. Algorithms for Solving Parity Games: Games on Graphs.
9. Solving Repeated Reachability and Eventual Safety Games.
10. Solving Parity Games.
11. Back and Forth Between Logic and Games: Introduction, Reachability Games and Parity Games, Reachability Games and Logic.
12. Logics with Least and Greatest Fixed-Points, Definability of Winning Regions in Parity Games, Inflationary Fixed-Point Logic and Backtracking Games.
13. Graph Searching Games: Introduction, Classifying Graph Searching Games, Variants of Graph Searching Games.
14. Monotonicity of Graph Searching, Obstructions, An Application to Graph-Decompositions, Complexity of Graph Searching.
15. Beyond Nash Equilibrium: Solution Concepts for the 21st Century: Robust and Resilient Equilibrium.
16. Taking Computation into Account, Taking (Lack Of) Awareness into Account, Iterated Regret Minimization.

Recommended Texts

1. Davis, M. D. (2012). Game theory: a nontechnical introduction. Chelmsford: Courier Corporation.
2. Apt, K. R., & Grädel, E. (Eds.). (2011). Lectures in game theory for computer scientists. Cambridge: Cambridge University Press.

Suggested Readings

1. Hearn, R. A., & Demaine, E. D. (2009). *Games, puzzles, and computation*. Boca Raton: CRC Press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-7419 Grammar Engineering 3 (3+0)

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview, LFG Basics.
3. LFG Basics II, Templates I, MacOSX, Unix.
4. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file
5. Lexical Rules, Passive and Argument alternations.
6. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.
11. Finite-State Morphology (FSM) I.
12. FSM II (-unknown), Free Word Order and the Shuffle Operator.
13. Meta-categories, Meta-rule-macros and Coordination.
14. Project

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. Stanford: CSLI. Publications Stanford.
2. Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., & Newman, P. (2008). *XLE documentation.* California: Palo Alto Research Center.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar.* Leiden: Brill.

2. Dalrymple, M., Kaplan, R. M., Maxwell III, J. T., Maxwell, J. C., & Zaenen, A. E. (1995). *Formal issues in lexical-functional grammar* (No. 47). Stanford: Center for the Study of Language (CSLI).

CSEC-7420 Graph Theory & Algorithms 3 (3+0)

This course is aimed to cover a variety of different problems in Graph Theory. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Various graphs algorithms will also be taught along with its analysis. By taking this course, one would be able to master fundamental concepts in Graph Theory, get to know a wide range of different Graphs, and their properties, be able to perform Elementary, Advanced Operations on Graphs to produce a new Graph, understand Graph Coloring, understand Eulerian and Hamiltonian paths and circuits. And many related topics to Paths, know how to turn a Graph into a Matrix and vice versa, obtain a solid foundation in Trees, Tree Traversals, and Expression Trees, have a good understanding of Graph Match. Topics covered in this course include: Fundamental Concepts of Graphs, Sub-Graphs and Super-Graphs, Connected Graphs (Walks and connection), Trees (including forests, spanning trees, Calay's Formula for positive integers, and other applications of trees), Non-Separable graphs and its applications, Connectivity among graphs, Planar graphs, Vertex coloring, edge coloring, Hamiltonian Cycles (including both Hamiltonian and non-Hamiltonian Graphs), Eigen values of graphs, Covering and packings.

Contents

1. Fundamental Concepts of Graphs: What is A Graph, Simple Graphs, Graph and Their Representations, Isomorphism and Automorrphisms, Labelled Graphs, Graphs Arising from Other Structures, Incidents Graphs, Union and Intersection Graphs Cartesian Product, Directed Graphs.
2. Sub-Graphs: Sub-Graphs and Super graphs, Spanning and Induced Sub-Graphs, Decomposition and Coverings, Edge Cuts and Bonds, Even Sub-Graphs, Graph Reconstruction.
3. Connected Graphs: Walks and Connection, Cut Edges, Connection to Diagraphs, Cycle Double Covers.
4. Trees: Forests and Trees, Spanning Tree, Calay's Formula, Fundamental Cycles AndBonds, Co-Tree, Trees and Distance. Applications of Tree.
5. Non-separable Graphs: Cut Vertices, Separations and Blocks, Ear Decompositions, Strong Orientations, Directed Ear Decompositions, Even Cycles Decompositions.
6. Connectivity: Vertex Connectivity, Fan Lemma, Edge Connectivity, Three-Connected Graphs, Submodularity, Determining, ChordalGraphs, Simplicial vertices.
7. Planar Graphs: Plane and Planar Graphs, Duality, Euler's Formula, Bridges.
8. Vertex Colorings: Chromatic Numbers, Critical Graphs, Girthand Chromatic Number.
9. Edge Colorings: Edge Colouring Number, Vizing's Theorem, Snarks.
10. Hamilton Cycles: Hamiltonian and non-Hamiltonian graphs.
11. Eigenvalues of Graph,
12. Coverings and Packings in Directed Graphs.
13. Integer Flows and Coverings.

Recommended Texts

1. Chartrand, G., & Zhang, P. (2013). *A first course in graph theory*. Chelmsford: Courier Corporation.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Chartrand, G., Lesniak, L., & Zhang, P. (2010). *Graphs & digraphs* (Vol. 39). Boca Raton: CRC press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-7421 Information Retrieval 3 (3+0)

Information retrieval covers the tasks of indexing, searching, and recalling data, particularly text or other unstructured forms. It has an important role to play in a large number of applications viz., digital libraries, office automation, internet and e-commerce. The aim of the course is to study theoretical aspects as well as implementation issues of classical and modern retrieval problems. Web search is the application of information retrieval techniques to the largest corpus of text anywhere — the web — and it is the context where many people interact with IR systems most frequently. In this course, we will cover basic and advanced techniques for building text-based information systems, including the following topics: Efficient text indexing, Boolean and vector-space retrieval models, Evaluation and interface issues, IR techniques for the web, including crawling, link-based algorithms, and metadata usage, Document clustering and classification, Traditional and machine learning-based ranking approaches. Course will also include few practical exercises.

Contents

1. Introduction: Basic Concepts of IR, Data Retrieval & Information Retrieval, IR System Block, Diagram. Automatic Text Analysis, Luhn's Ideas, Conflation Algorithm, Indexing and Index Term Weighing, Probabilistic Indexing, Automatic, Classification. Measures of Association, Different Matching Coefficient, Classification Methods, Cluster Hypothesis. Clustering Algorithms.
2. File Structures, Inverted File, Suffix Trees & Suffix Arrays, Signature Files, Ring Structure, IR Models, Basic Concepts, Boolean Model, Vector Model, and Fuzzy Set Model. Search Strategies, Boolean Search, Serial Search, and Cluster based Retrieval, Matching Function
3. Performance Evaluation- Precision and Recall, Alternative Measures Reference Collection.
4. Taxonomy and Ontology: Creating Domain Specific Ontology, Ontology Life Cycle.
5. Multimedia IR Models & Languages- Data Modeling, Techniques to Represent Audio and Visual Document, Query Languages Indexing & Searching- Generic Multimedia Indexing Approach.
6. Searching the Web, Challenges, Characterizing the Web, Search Engines, Browsing.

Recommended Texts

1. Ricardo, B. Y., & Berthier, R. N. (2011). *Modern information retrieval: the concepts and technology behind search*. New Jersey, Boston: Addison-Wesley Professional.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Grossman, D. A., & Frieder, O. (2012). *Information retrieval: Algorithms and heuristics* (vol. 15). New York: Springer Science & Business Media.
2. Levene, M. (2011). *An introduction to search engines and web navigation*. Hoboken: John Wiley & Sons.
3. Subrahmanian, V. S., & Jajodia, S. (Eds.). (2012). *Multimedia database systems: issues and research directions*. New York: Springer Science & Business Media.

ITEC-7404 **Information Security** **3 (3+0)**

As IT continues to grow at a rapid pace and plays a significant role in automating the functions of the enterprise, Information Security has taken on an unparalleled significance around the world. This course is based on Introduction to Information Security: History of IS, what is Security, CNSS Security model, Component of Information Systems, Balancing Information Security and Access, Approaches to Information Security Implementation, The Security System Development Life Cycle, Security Professionals and the Organizations, Communities of Interest. Finally, students will discover how organizations manage and prepare for security incidents, disruptions and disasters and how they manage the day-to-day operations of an information security program. When you are finished with this course, you will have the knowledge and understanding of the bigger picture of information security. Students understand of various types of security incidents and attacks, and learn methods to prevent, detect and react incidents and attacks. Students will also learn basics of application of cryptography which are one of the key technologies to implement security functions.

Contents

1. The Need for Security: Business Need, Threats, Attacks, Secure Software Development. Legal, Ethical, and Professional Issues in Information Security: Laws and Ethics in Information Security, International Laws and Legal Bodies, Ethics and Information Security, Codes of Ethics and Professional Organizations, International Agencies.
2. Risk Management: Introduction to RM, Risk Identification, Risk assessment, Risk Control Strategies, Selecting a risk Control Strategy, Quantitative VS qualitative Risk Control Practices, Risk Control Practices.
3. Planning for Security: Information Security Planning and Governance, Information Security Policies, Standards, and Governance, Information Security Blue Prints, Continuity Strategies. Security Technology: Firewalls, VPNs, and Wireless: Access Control, Firewalls, Protecting Remote Connections. Security Technology: Intrusion Detection and Prevention Systems and Other Security Tools: Intrusion Detection and Presentation Systems, Honeypots, Honeynets, and Padded Cell Systems, Scanning and Analysis Tools, Biometric Access Control.
4. Cryptography: Foundations of Cryptography, Cipher Methods, Cryptographic Algorithms, Cryptography Tools, Protocols for secure Communications, Attack on Cryptosystems. Physical Security: Physical Access Controls, Fire Security and Safety, Failure of Supporting Utilities and Structural Collapse, Interception of Data, Considerations for Physical Security.
5. Implementing Information Security: Information Security Project Management, Technical Aspects of Implementation, Nontechnical Aspects of Implementation, Information Security Certifications.
6. Security and Personnel: Positioning and Staffing Function, Credentials of Information Security Professionals, Employment Policies and Practices, Security Considerations for Non-employees. Information Security Maintenance and eDiscovery: Security Management Maintenance Models

Recommended Texts

1. Michael E. Whitman and Herbert J. Mattord. (2011). *Principles of information security,* (4th ed.). San Francisco: Course Technology.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. James M. Stewart, Mike Chapple, and Darril Gibson. (2012). *Information systems security professional study guide*, (6th ed.). Hoboken: Sybex.
3. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-7422 **Information Theory** **3 (3+0)**

Information Theory is one of the few scientific fields fortunate enough to have an identifiable beginning - Claude Shannon's 1948 paper. The story of the evolution of how it progressed from a single theoretical paper to a broad field that has redefined our world is a fascinating one. It provides the opportunity to study the social, political, and technological interactions that have helped guide its development and define its trajectory, and gives us insight into how a new field evolves. The course will introduce students with concepts such as 1) Demonstrate knowledge and understanding of the fundamentals of information theory. 2) Appreciate the notion of fundamental limits in communication systems and more generally all systems. 3) Develop deeper understanding of communication systems. 4) Apply the concepts of information theory to various disciplines in information science, 5) understand entropy and entropy Ergodic theorem, 6) understand distortion and approximation techniques. Course will also include few practical exercises.

Contents

1. Information Sources
2. Pair Processes
3. Block Independent Channels, Conditionally Block Independent Channels, Stationarizing Block Independent Channels, Primitive Channels, Additive Noise Channels, Markov Channels
4. Finite-State Channels and Codes, Cascade Channels, Communication Systems, Couplings, Block to Sliding-Block: The Rohlin-Kakutani Theorem.
5. Entropy
6. The Entropy Ergodic Theorem
7. Distortion and Approximation
8. Approximating Random Vectors and Processes, The Monge/Kantorovich/Vasershtein Distance.
9. Variation and Distribution Distance, Coupling Discrete Spaces with the Hamming Distance, Process Distance and Approximation, Source Approximation and Codes, d-bar Continuous Channels.
10. Distortion and Entropy
11. Relative Entropy
12. Information Rates
13. Distortion and Information
14. Ergodic Theorems for Densities
15. Source Coding Theorems
16. Asynchronous Block Codes, Sliding-Block Source Codes, Geometric Interpretation, Properties of Good Source Codes, Optimal and Asymptotically Optimal Codes, Block Codes Sliding-Block Codes.

Recommended texts

1. Robert M. Gray. (2011). *Entropy and information theory,* (2nd ed.). New York: Springer.
2. John Scales Avery. (2012). *Information theory and evolution,* (2nd ed.). Bukit Batok: World Scientific Publishing Company.

Suggested Readings

1. Thomas M. Cover, Joy A. Thomas. (2006). *Elements of information theory,* (2nd ed.). Hoboken: Wiley-Interscience.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-7423 Intelligent Multi Agent Systems 3 (3+0)

A multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents. MAS can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may include methodic, functional, procedural, approaches, algorithm-ic, search or reinforcement learning. Intelligent MAS is an important area having multiple applications in Computing. They are a useful computational paradigm for creating systems that are flexible, adapt to change of the environment, and are able to integrate heterogeneous components. To address these characteristics, a number of issues are considered when studying MAS: how can an agent communicate and coordinate its activities with other agents in the system, how can agents represent and reason about the state of their interaction process, how can they represent and reason about actions, plans and knowledge of other agents, decompose goals and distribute tasks, what architecture can they be given so that they can solve a particular problem.

Contents

1. Intelligent Agents
2. Multi agent Systems
3. Agent Interaction Protocols
4. Distributed Problem Solving & Planning
5. Search Algorithm for Agents
6. Distributed Rational Decision Making
7. Learning in Multi agent Systems
8. Computational Organization Theory
9. Formal Methods in DAI (Distributed Artificial Intelligence)
10. Industrial and Practical Applications of DAI
11. Groupware and Computer Supported Cooperative Work
12. Distributed Models for Decision Support
13. Concurrent Programming for DAI

Recommended Texts

1. Gerhard Weiss. (2000). Multiagent systems: *A modern approach to distributed artificial intelligence* Cambridge: The MIT Press.
2. Tomas Salamon. (2011). *Design of agent-based models.* Brno: Eva & Tomas Bruckner.

Suggested Readings

1. Maria Fasli. (2007). *Agent technology for e-Commerce*, (1st ed.). Hoboken: Wiley.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

CSEC-7424 **Intelligent Agents Technology** **3 (3+0)**

The course focuses on the use of Intelligent Agent Technologies for development of complex software systems driven by collective intelligence. Intelligent software agents are such self-managed (autonomic) software entities that are capable to carry out some goal-driven and knowledge-based behavioral activities on behalf of a user or some other software application, which created it. This theory-oriented part of the course reviews appropriate AI methods and technologies needed to enable intelligent agents. After passing this course student will be able to understand about software agent technology including: What is agent technology? Technological contexts, Trends and drivers, Agent technologies and tools, Applications, Challenges and Technology roadmaps. The course focuses on the use of Distributed Artificial Intelligence methods, and more specifically of Intelligent Agents Technologies, for development of complex distributed software systems driven by collective intelligence. This theory-oriented part of the course reviews appropriate AI methods and technologies needed to enable intelligent agents.

Contents

1. Intelligent Agents: Attributes, Task level Skills, Knowledge, Communication Skills, End user Taxonomy of Agents, Types of Agents
2. Desktop Agents: Operating System Agents - An Intelligent Utility System Agent, Interface Agent, Application Agents, Email Agent, Search Agent, Application Suite Agents.
3. Internet Agents: Web search Agents, Benefits of Search Agents, Information Filtering Agents and its Benefits, Offline Delivery Agents and its Benefits, Notification Agents.
4. Intranet Agents: Intranet Search Agents, Informational Filtering Agents, Collaboration Agents, Lotus Notes Agents, Process Automation Agents – Edify work force.
5. Technology of Intelligent Agents: the historical view, technical view, Intelligence & Agency, Machinery, Content, Access & Security, Development Model, Steps towards Automation.
6. Agent Machinery: Principles of Reasoning – Inferencing Systems, Rules of Inferencing, Sensors & Actors, RAISE: An Inferencing System for Agents, The Trouble with Rules.
7. Agent Content: Knowledge Representation, Agent Communication, World Views –World views for Agents, Classes of World-Views, World Views Models for your business.
8. Agent Access: A Framework for Intelligent Agent Access, Agent Access for Existing Applications –Wrappering, Client/Server Applications, Database & File System Interfaces.
9. Agent Security: What does Security Do? Security for Agents, Security Basics, Delegation for Agents, Limiting Resource Consumption.
10. Developing Agent Applications: Learning Agent Functional Architecture.

Recommended Texts

1. Salamon, T. (2011). *Design of agent-based models*. Brno: Eva & Tomas Bruckner Publishing.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Readings

1. Caglayan, A., & Harrison, C. (1997). *Agent sourcebook*. Hoboken: John Wiley & Sons, Inc.
2. Weiss, G. (Ed.). (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge: The MIT press.
3. Brenner, W., Zarnekow, R., & Wittig, H. (2012). *Intelligent software agents: foundations and applications*. New York: Springer Science & Business Media
4. Fasli, M. (2007). *Agent technology for e-commerce* (vol. 3). Hoboken: John Wiley & Sons.

ITEC-7405 Knowledge Management Systems & Technologies 3 (3+0)

Knowledge Management (KM), is an integrated interdisciplinary presentation that makes sense of the confusingly wide variety of computer science and business KM perspectives arising simultaneously from artificial intelligence, information systems, and organizational behavior. This course examines the characteristics and applications of systems that support knowledge management at personal, group and corporate levels. It considers the electronic representation of knowledge, the components of systems that embody or support knowledge processing and the use and value of such systems. The issue of knowledge management systems has probably always been the most discussed and debated topic within knowledge management. This course will discuss the theoretical implementation of knowledge management systems and its impact on the organization. By the end of the course, students should be able to: Appraise current thought on knowledge management in the light of contemporary debates on knowledge productivity, strategic capability and organizational learning ;Apply theories of knowledge management relevant to current workplace practice; Apply the tools and techniques of knowledge management.

Contents

1. History and paradigms of knowledge management;
2. Types of knowledge,
3. Knowledge Economy,
4. Knowledge Management.
5. KM Processes
6. KM Frameworks and Models
7. KM Frameworks and Models
8. Knowledge Capture and Codification
9. Knowledge Sharing and Communities of Practice
10. Knowledge Application
11. The Role of Organizational Culture
12. Knowledge Management Tools
13. Knowledge Management Strategy
14. The Value of Knowledge Management
15. Organizational Learning and Organizational Memory
16. The KM Team

Recommended Texts

1. KimizDalkir. (2011). *Knowledge management in theory and practice*, (3rd ed.). Cambridge: The MIT Press.
2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

Suggested Reading

1. Elie Geisler and Nilmini Wickramasinghe (January 15, 2009) *Principles of knowledge management: theory, practice and cases*. New York: M.E. Sharpe.
2. Murray E. Jennex (August 10, 2007) *Knowledge management: concepts, methodologies, tools and applications* (6-volume set). Hershey: IGI Global.

CSEC-7425 Machine Learning 3 (3+0)

In this course, student will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work. More importantly, student will learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. The students will learn basic concepts and techniques of Machine Learning. They can develop skills of using recent machine learning software for solving practical problems. Students will gain experience of doing independent study and research. More specifically, it covers the following topics: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that a student could also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

Contents

1. Introduction to machine learning.
2. Linear regression.
3. Gradient Descent as general parameter Learning/optimization approach.
4. Logistic regression, One-vs-all classification, Regularization.
5. Neural Networks.
6. Practical advice for applying learning algorithms: How to develop, debugging, feature/model design, setting up experiment structure
7. Probability and Bayesian Analytics.
8. Naive Bayes.
9. Support Vector Machines (SVMs) and the intuition behind them.
10. Unsupervised learning: clustering and dimensionality reduction.
11. Anomaly detection.
12. Probabilistic Graphical Models: Bayesian Belief Networks.
13. Probabilistic Graphical Models: Learning from Data.
14. Probabilistic Graphical Models: EM algorithm.
15. Hidden Markov Models.

Recommended Texts

1. Christopher M. Bishop. (2006). *Pattern recognition and machine learning*. New York: Springer.
2. Tom Mitchell. (1997). *Machine learning*. New York: McGraw-Hill.

Suggested Readings

1. Trevor Hastie, Robert Tibshirani and Jerome Friedman. (2009). *The elements of statistical learning*. New York: Springer.

2. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.

ITEC-7406 **Multimedia Databases** **3 (3+0)**

The spatial, temporal, storage, retrieval, integration and presentation requirements of multimedia data differ significantly from those for traditional data. A multimedia database management system provides for the efficient storage and manipulation of multimedia data in all its varied forms. We look into the basic nature of multimedia data, highlight the need for multimedia DBMSs, and discuss the requirements and issues necessary for developing such systems. Querying a multimedia database consists of retrieving text, images, video, and audio simultaneously. Because the data in a multimedia database are usually not as structured as the data in a conventional database, querying multimedia data is not a trivial problem and has been a popular research area. This course aims to provide a basic study of the development of fundamental multimedia database systems, as well as applicable technologies for developing web-based multimedia applications. The former provides a basis for understanding the basic concepts and techniques pertinent to multimedia databases. In this course, we examine the aspects regarding building multimedia database systems and give an insight into the used techniques. The course deals with content-specific retrieval of multimedia data.

Contents

1. Introduction to Multimedia Databases
2. Multimedia Data
3. An Introduction to SQL and Multimedia
4. Querying Multimedia Data
5. Multimedia Database Architecture and Performance
6. Dealing with Text Databases
7. Dealing with Video Databases.

Recommended Texts

1. Dunckley, L. (2003). Multimedia databases: *An object relational approach*. Boston: Addison-Wesley Longman Publishing Co., Inc.
2. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

Suggested Readings

1. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.
2. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

CSEC-7426 Multimedia Systems Theory & Principles 3 (3+0)

The rapid growth of multimedia technology has made it possible to deliver high quality audio, graphics, video and animation to the user. However, this growth in technology has not been met by a growth in design knowledge. While it is possible to have multimedia, it is not at all obvious that we know how to design high-quality multimedia systems that are fully usable to the degree we should expect. The course introduces students with theory and principles of multimedia contents constituting multimedia system. Different topics are given as a term paper. Extensive literature review is the part of this. Topics covered for term papers are Compression algorithms including Lossless compression, Image compression, Lossy compression, video compression techniques, audio compression techniques and MPEG compression. A prototype based on compression technique would be developed by students. Different Graphs and image representation schemes are also part of this course. Digital interfacing and video interfacing are also part of the course.

Contents

1. Introduction to Multimedia: What is Multimedia? Multimedia and Hypermedia, Overview of Multimedia Software Tools.
2. Graphics and Image Data Representations: Graphics/Image Data Types, Popular File Formats
3. Color in Image and Video: Color Science, Color Models in Images, Color Models in Video Fundamental Concepts in Video: Types of Video Signals, Analog Video, Digital Video.
4. Basics of Digital Audio: Digitization of Sound, MIDI: Musical Instrument Digital Interface, Quantization and Transmission of Audio.
5. Lossless Compression Algorithms: Introduction, Basics of Information Theory, Run-Length Coding, Variable-Length Coding (VLC), Dictionary-Based Coding, Arithmetic Coding, Lossless Image Compression.
6. Lossy Compression Algorithms: Distortion Measures, The Rate-Distortion Theory, Quantization, Transform Coding, Wavelet-Based Coding, Wavelet Packets.
7. Image Compression Standards: The JPEG Standard, The JPEG2000 Standard, The JPEG-LS Standard
8. Basic Video Compression Techniques: Introduction to Video Compression, Video Compression Based on Motion Compensation, Search for Motion Vectors, H.261, H.263.
9. MPEG Video Coding MPEG-1 and 2.
10. MPEG Video Coding MPEG-4, 7.
11. Basic Audio Compression Techniques: ADPCM in Speech Coding, G.726 ADPCM, Vocoders.
12. MPEG Audio Compression: Psychoacoustics, MPEG Audio.

Recommended Texts

1. Z. M. Li and M. S. Drew. (2004). *Fundamentals of multimedia.* New York: Prentice Hall.
2. by N. Chapman and J. Chapman. (2004). *Digital multimedia,* (2nd ed.). Hoboken: Wiley.

Suggested Readings

1. Frank Y. Shih (2012) *Multimedia security: watermarking, steganography, and forensics,* (1st ed.). Boca Raton: CRC Press.
2. Daniel Cunliffe and Geoff Elliott (2005) *Multimedia computing*. Essex: Lexden Publishing Ltd

CSEC-7427 **Natural Language Processing** **3 (3+0)**

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes, Context-free parsing, Earley algorithm, Extending CFG.
4. Probabilistic parsing, Parsing tricks, Human sentence processing.
5. Semantics, Forward-backward algorithm, Expectation Maximization.
6. Finite-state algebra, Finite-state implementation, Finite-state tagging, Noisy channels and FSTs, More FST examples.
7. Programming with regexps, Morphology and phonology.
8. Optimal paths in graphs, Structured prediction.
9. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Recommended Texts

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. Sebastopol: OReilly Media.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Bengio, Y. (2007). *Learning deep architectures for AI*.
4. Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge: MIT press

CSEC-7428 Parallel Processing: Algorithms and Architectures-I 3 (3+0)

The course covers the needed perspective, taxonomy, and analysis tools for the understanding key concepts of parallel processing. It also delimits the models of parallel processing which has become quite important in recent years. More specifically, it deals with parallelism, its needs, type and its importance along with some simple architecture for better illustration, complexity of parallel algorithms (including asymptomatic, optimality and efficiency), PRAM algorithm along with its sub models and assumptions, selection and sorting networks and its design, Sorting and routing on a 2-D mesh, and other mesh related architectures. Paralleling processing covers Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory. Other Mesh-Related Architectures: Three or more dimensions, Stronger and weaker connectivity, meshes augmented with non-local links, meshes with dynamic links, Pyramid and multi-grid systems, Meshes of trees.

Contents

1. Introduction to Parallelism: Why parallel processing? Parallel processing ups and downs, Types of parallelism: A taxonomy, Constraints of parallel processing, Effectiveness of parallel processing. Introduction to Parallel Algorithms: Some simple computations, some simple architectures, Algorithms for a linear array, Algorithms for a binary tree.
2. Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory.
3. PRAM and Basic Algorithms: PRAM sub models and assumptions, Data broadcasting, Semigroup or fan-in computation, Parallel prefix computation, Ranking the elements of a linked list
4. Sorting and Selection Networks: What is a sorting network? Figures of merit for sorting networks, Design of sorting networks, Batcher sorting networks
5. Sorting on a 2-D Mesh or Torus: Mesh-connected computers, Theshearsort algorithm.
6. Routing on a 2-D Mesh or Torus: Types of data routing, Useful elementary operation
7. Numerical 2-D Mesh Algorithms: Matrix multiplication, Triangular system of equations, Tridiagonal system, Arbitrary system of linear equations, Graph algorithms, Image-processing algorithms
8. Other Mesh-Related Architectures: Three or more dimensions, Stronger and weaker connectivity, meshes augmented with non-local links, Meshes with dynamic links.

Recommended Texts

1. Behrooz Parhami. (2006). *Introduction to parallel processing: algorithms and architectures*, New York: Springer.
2. Michel Cosnard and Denis Trystram (September 1996) *Parallel algorithms and architectures*, London: Intl Thomson Computer Pr (Sd).

Suggested Readings

1. Seyed H Roosta. (1999). *Parallel processing and parallel algorithms: theory and computation*, (1st ed.). New York: Springer.
2. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

CSEC-7429 Parallel Processing: Algorithms and Architectures-II 3 (3+0)

The course covers the needed perspective, taxonomy, and analysis tools for the understanding key concepts of parallel processing. It also delimits the models of parallel processing which has become quite important in recent years. More specifically, it deals with parallelism, its needs, type and its importance along with some simple architecture for better illustration, complexity of parallel algorithms (including asymptomatic, optimality and efficiency), PRAM algorithm along with its sub models and assumptions, selection and sorting networks and its design, Sorting and routing on a 2-D mesh, and other mesh related architectures. Paralleling processing covers Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory. Other Mesh-Related Architectures: Three or more dimensions, Stronger and weaker connectivity, meshes augmented with non-local links, meshes with dynamic links, Pyramid and multi-grid systems, Meshes of trees.

Contents

1. Hypercubes and Their Algorithms: Definition and main properties, Embeddings and their usefulness, embedding of arrays and trees, A few simple algorithms, Matrix multiplication, Inverting a lower triangular matrix.
2. Sorting and Routing on Hypercubes: Defining the sorting problem, Bitonic sorting, Routing problems on a hypercube, Dimension-order routing, Broadcasting on a hypercube, Adaptive and fault-tolerant routing. Other Hypercubic Architectures: Modified and generalized hypercubes, Butterfly and permutation networks, Plus-or-minus-2i network, The cube-connected cycles, Shuffle and shuffle-exchange networks.
3. Sampler of Other Networks: Performance parameters for networks, Star and pancake networks, Ring-based networks, Composite or hybrid networks, Hierarchical (multi-level) networks
4. Data Storage, Input, and Output: Data access problems and caching, Cache coherence protocols
5. Reliable Parallel Processing: Defects, faults, failures, Defect-level methods, Fault-level methods, Error-level methods, Malfunction-level methods, Degradation-level methods
6. Shared-Memory MIMD Machines: Variations in shared memory, MIN-based BBN Butterfly, Vector-parallel Cray Y-MP, Latency-tolerant Tera MTA, CC-NUMA Stanford DASH
7. Data-Parallel SIMD Machines: Where have all the SIMDs gone? The first supercomputer: ILLIAC IV, Massively parallel Goodyear MPP, Distributed Array Processor (DAP).

Pre-Requisite: Parallel Processing: Algorithms and Architectures-I

Recommended Texts

1. BehroozParhami. (2006). *Introduction to parallel processing*: *algorithms and architectures*. New York: Springer.
2. Michel Cosnard and Denis Trystram. (1996). *Parallel algorithms and architectures,* London: Intl Thomson Computer Pr (Sd).

Suggested Readings

1. Seyed H Roosta (December 10, 1999) *Parallel processing and parallel algorithms: theory and computation,* (1st ed.). New York: Springer.

SEEC-7401 **Reliability Engineering** **3 (3+0)**

One of the most difficult issues in developing a system product is determining how long it will last and how long it should last. Reliability describes the ability of a system or component to function under stated conditions for a specified period of time. Reliability is closely related to availability, which is typically described as the ability of a component or system to function at a specified moment or interval of time. This idea of reliability that was birthed in the early days of electronics and aviation, now extends into every sector of consumer and industrial products. Automobiles, airplane, televisions and computers have all found their way into the hands of everyday consumers because of the advancement in reliability engineering. The main objectives of reliability engineering are: To apply engineering knowledge and specialist techniques to prevent or to reduce the likelihood or frequency of failures. To identify and correct the causes of failures that occur despite the efforts to prevent them.

Contents

1. Introduction and Overview, LFG Basics.
2. LFG Basics II, Templates I, MacOSX, Unix.
3. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file.
4. Lexical Rules, Passive and Argument alternations.
5. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic
6. Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation.
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.
11. Finite-State Morphology (FSM) I.
12. FSM II (-unknown), Free Word Order and the Shuffle Operator.
13. Meta-cateogries, Metarulemacros and Coordination.
14. Project

.

Recommended Texts

1. Ebeling, C. E. (2009). *An introduction to reliability and maintainability engineering*, (2^{nd} ed.). Illinois: Waveland Press, Inc.

Suggested Readings

1. by Michael R. Lyu. (2008). *IEEE recommended practice in software reliability handbook of software reliability engineering,* New York: McGraw-Hill Book Company,

SEEC-7402 **Research Methodology** **3 (3+0)**

Research can be defined as “an activity that involves finding out, in a more or less systematic way, things you did not know. Methodology is the philosophical framework within which the research is conducted or the foundation upon which the research is based. Students will understand research terminology, research objectives and importance. They will become aware of the ethical principles of research, ethical challenges, approval processes and types of researches. It will provide an opportunity for participants to establish or advance their understanding of research through critical exploration of research language, ethics, and approaches. The course introduces the language of research, ethical principles and challenges, and the elements of the research process within quantitative, qualitative, and mixed methods approaches. Participants will use these theoretical underpinnings to begin to critically review literature relevant to their field or interests and determine how research findings are useful in forming their understanding of their work, social, local and global environment.

Contents

1. Introduction to Research. Objectives of Research. Importance of Research Methodology in Research Study.
2. Types of Research. Steps in Conducting Research.
3. What is Literature Review? Why need for Literature Review. Types of Literature Review. Systematic Literature Review Protocol.
4. Problem Statement and Problem formulation. Criteria for selecting a problem.
5. Identifying Types of variables in Research.
6. Types of hypothesis. Identifying Target Population. Types of Sampling. Sampling Techniques.
7. Quantitative Research Methods.
8. Scientific Methods. Design of Quantitative Surveys. Techniques to Conduct Quantitative Methods.
9. Introduction to Qualitative Research. Qualitative Research Methods.
10. Data Analysis and Theory in Qualitative Research Articles
11. Introduction to Mixed Methods Research. Design of Mixed Methods Research. Evaluation of Mixed Methods Research.
12. Case Study. How to Conduct a Case Study. Case Study Protocol.
13. Importance and Benefits of Case Study. Types of Statistical Tests to Conduct Data Analysis. Data
14. Analysis Tools.
15. Introduction to SPSS. Hands on Practice of SPSS.
16. How to Define variables in SPSS. How to Record Collected Data in SPSS.
17. Types of Tests via SPSS including Regression.
18. Correlation. Cross tabulation and others.
19. How to write Good Research Proposal.
20. Contents of Thesis.
21. Important Elements of Research Thesis.

.

Recommended Texts:

1. Creswell, J. W. (2014*). Research design: Qualitative, quantitative and mixed methods approaches*, (2nd ed.). Thousand Oaks: SAGE.

ITEC-7407 **Service-Oriented Architecture** **3 (3+0)**

Service-oriented architecture (SOA) emerged in the early part of this century as an evolution of distributed computing. Before SOA, *services* were understood as the end result of the application development process. Before SOA emerged in the late 1990s, connecting an application to data or functionality housed in another system required complex point-to-point integration—integration that developers had to recreate, in part or whole, for each new development project. Exposing those functions through SOA eliminates the need to recreate the deep integration every time. In SOA, the application itself is composed of services. Services can be delivered individually or combined as components in a larger, composite service. By the end of this course students will be able to understand SOA, Service Orientation and Web Services. They will also be able to analyze and design business based on SOA principles. After successful completion of course student will be able to describe SOA (Service-Oriented Architecture) to structure web-based systems, explain WS* services (i.e., SOAP over HTTP, WSDL, UDDI, BPEL), apply REST architecture (i.e., JSON over HTTP, URI) and identify REST design principles.

Contents

1. Introducing SOA
2. Web Services & Primitive SOA
3. Web Services & Contemporary SOA (Activity Management & Composition)
4. Principles of Service-Orientation
5. Service Orientation & Contemporary SOA
6. SOA Delivery Strategies SOA Delivery Lifecycle Phases, Top Down Strategy, Bottom Up Strategy, Agile Strategy.Service Oriented Analysis: Introduction, Benefits of a Business Centric SOA, Deriving Business Services, Deriving Business Services.
7. Service Modelling (Process): Introduction, Modelling Guidelines, Classifying Service Model Logic, Different Modelling Approaches.
8. Service Oriented Design: Introduction, WSDL-Related XML Schema Language, WSDL Language(Like Definitions, Types, Message & Part, Porttype, Interface, & Operation Elements, Input & Output-When Used With Operation, Binding, Input & Output-When Used With Binding, Service, Port & Endpoint, Import, Documentation), SOAP Language Basic (Envelop, Header, Body, Fault) Service Interface Design Tools.
9. Service Oriented Design Steps To Composing SOA, Considerations For Choosing Service Layers, Considerations For Positioning Core SOA Standards (Industry Standards, & SOA, XML & SOA, The WS-I Baisc Profile, WDSL & SOA, XML Schema & SOA, SOAP & SOA, Namespaces & SOA, UDDI & SOA) Considerations For Choosing SOA Extensions (Choosing SOA Characteristics, Choosing WS* Specifications, WS-BPEL & SOA).
10. Service Oriented Design: Overview, Entity Centric Business Service Design, Application Service Design, Service Design Guidelines.

.

Recommended Texts

1. Erl, T. (2005) *Service-oriented architecture (SOA): concepts, technology, and design*, New York: Pearson Education Incorporated.

CSEC-7430 **Soft Computing** **3 (3+0)**

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. It is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modeling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc. More specifically, it is collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low-cost solutions. It is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect, the role model for soft computing is the human mind.

Contents

1. Overview of Soft Computing,
2. Probabilistic Graphical Models,
3. Bayesian Networks,
4. Inferencing in Graphical Models,
5. Swarm Intelligence,
6. Ant Colony Optimization, Particle Swarm Optimization,
7. Cat Swarm Optimization,
8. Convergence Analysis,
9. Neural Networks,
10. Deep Learning,
11. Convolutional Neural Networks,
12. Recurrent Neural Networks
13. Relational Learning,
14. Tensor Factorization,
15. Ensemble Methods.

Recommended Books

1. Engelbrecht, A. P. (2010, September). *Heterogeneous particle swarm optimization. In international conference on swarm intelligence* (pp. 191-202). New York: Springer.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT press.

Suggested Readings

1. Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* Cambridge: MIT press.

SEEC-7403 **Software Configuration Management** **3 (3+0)**

This course provides a basic introduction to the theory, principles, and techniques of Software Configuration Management (SCM) as it applies to the entire software lifecycle. It addresses the application of SCM in a wide variety of approaches to software development and maintenance, from traditional to agile. The course illustrates the SCM strategies, techniques, and required tool capabilities that support each of the activities in the software development life cycle. The student will also gain a value-based understanding of which SCM techniques are most useful for the development approach and tool capabilities that currently exist in their company. The course concludes by providing an SCM implementation framework for planning and selecting the optimal SCM strategy and tool capabilities for both the project and organizational levels. More specifically, it deals with Management of the SCM Process, Constraints and Guidance for the SCM Process, Surveillance of SCM, Software Configuration Identification, SCM tools and current research in SCM.

Contents

1. Management of the SCM Process
2. Organizational Context for SCM
3. Constraints and Guidance for the SCM Process
4. Planning for SCM. SCM Plan
5. Surveillance of Software Configuration Management
6. Software Configuration Identification
7. Identifying Items to Be Controlled
8. Software Library
9. Software Configuration Control
10. Requesting, Evaluating, and Approving Software Changes
11. Implementing Software Changes
12. Deviations and Waivers
13. Software Configuration Status Accounting
14. Software Configuration Status Information
15. Software Configuration Status Reporting. Software Configuration Auditing
16. Software Functional Configuration Audit. Software Physical Configuration Audit
17. In-process Audits of a Software Baseline
18. Software Release Management and Delivery. Software Building
19. Software Release Management
20. Software Configuration Management Tools. Current research topics in Software Configuration Management

Recommended Texts

1. Tephen P. Berczuk, Brad Appleton. (2003). *Software configuration management patterns: effective teamwork, practical integration*. Boston: Addison Wesley.

SEEC-7404 Software Measurement and Metrics 3 (3+0)

A measurement is a manifestation of the size, quantity, amount or dimension of a particular attributes of a product or process. Software measurement is a titrate impute of a characteristic of a software product or the software process. It is an authority within software engineering. Software is measured to create the quality of the current product or process, anticipate future qualities of the product or process, enhance the quality of a product or process and regulate the state of the project in relation to budget and schedule. Measurement is critical for the successful management of software projects. It is the basis of project planning because it is used to establish achievable project targets. Measurement helps to monitor the progress of a project. A project is considered successful if it has met its targets in terms of the cost, the schedule, and the quality. It is possible to evaluate these by measuring the project-related data. The overall objective of this course is to determine adjusted function point count for a software system.

Contents

1. Introduction to quality control and planning needs
2. Measurement Concepts
3. Measurement as a support process
4. Review Metrics Models and Standards
5. Measurement goals
6. Formulating problem and goal statement
7. Prioritize information needs and objectives
8. Formalize measurement goals
9. Specify Measures
10. Identify questions and indicators
11. Identify data elements, Operational definitions for measures.
12. Specify Data Collection and Storage Procedures.
13. Sources of data. How to collect and store the measurement data?
14. Specify Analysis Procedures. Potential data analyses.
15. Methods and tools for measuring software.
16. Develop software measurement reporting.
17. Current research topics in Software Measurement and Metrics.

.

Recommended Texts

1. Stephen H. Kan. (2003). Metrics and models in software quality engineering. Boston: Addison Wesley.
2. Chris Chen and Hadley Roth. (2005). The big book of six sigma training games. New York: McGraw-Hill.

Suggested Readings

1. Anita Carleton, William A. Flora. (1999). Measuring the software process. Boston: Addison-Wesley

SEEC-7405 Software Risk Management 3 (3+0)

Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. A possibility of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor-quality software, not being able to complete the project on time. This course focuses on intermediate and advanced strategies you can use to manage general risks and details practical techniques you can use to control your project's specific risks. More specifically, this course is to introduce you to risk management, risk management paradigm, risk identification and its management in CMM, risk taxonomy, various tools for identifying the risks, classifying risks, its analysis, planning, and monitoring, best practices and lessons learned on Risks, various other Case study based exercises and presentations on approach by each team.

Contents

1. What is risk and risk management?
2. Motivation for risk management.
3. Reasons we don't do risk management.
4. SEI's Risk Management paradigm.
5. Identifying and recording software risk.
6. Risk Taxonomy.
7. Tools and methods for identifying and recording risks.
8. Analysing and classifying risks.
9. Complex project management theory.
10. Software Risk Identification.
11. Software Risk Analysis.
12. Software Risk Planning.
13. Software Risk Monitoring.
14. Software Qualitative Risk Analysis.
15. Quantitative Risk Analysis.
16. Risk management and the SDLC.
17. Risk management in CMM.
18. Other useful tools for successful risk management.
19. Current research topics in Software Risk Management.

Recommended Texts

1. Dale Walter Karolak. (1995). *Software engineering risk management.* Hoboken: Wiley.

Suggested Readings

1. C. Ravindranath Pandian. (2006). *Applied software risk management: A guide for software project managers*. Boca Raton: Auerbach Publications

CSEC-7431 **Speech Recognition and Synthesis** **3 (3+0)**

Speech Processing offers a practical and theoretical understanding of how human speech can be processed by computers. It covers speech recognition, speech synthesis and spoken dialog systems. The course involves practicals where the student will build working speech recognition systems, build their own synthetic voice and build a complete telephone spoken dialog system. This work will be based on existing toolkits. Details of algorithms, techniques and limitations of state-of-the-art speech systems will also be presented. This course is designed for students wishing understand how to process real data for real applications, applying statistical and machine learning techniques as well as working with limitations in the technology. This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background:
2. N-gram Language Modelling.
3. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization J+M: Phonetics , J+M Speech Synthesis, J+M Word Classes and Part-of-Speech Tagging, but skip section 5.6 , Optional Advanced Reading: Chapter 4 "Text Segmentation and Organisation "Text Decoding" .
4. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software J+M, Read sections 1, 2, 3, 4, 5, and 6.1 and 6.1.1 from Alan Black's lecture notes on TTS and Festival. , Festival manual (used as Festival's scripting language): Introduction to Scheme for C Programmers, from Cal Tech. Optional Advanced Reading: "Prosody Prediction from Text" from Taylor, Paul. 2007.
5. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis) J+M New, pages 25-end Optional Advanced Reading:"Unit Selection Synthesis" from Taylor, Paul. 2007.
6. Optional Advanced Reading: Optional: Section 7 from Alan Black's lecture notes on TTS and Festival. Optional Advanced Reading: The rest of Section 6 of Alan Black's lecture notes
7. ASR: Noisy Channel Model, Bayes
8. HMMs, Forward, Viterbi J+M: Hidden Markov Models, pages 1-20 J+M
9. ASR: Feature Extraction and Acoustic Modelling, Evaluation J+M: Automatic Speech Recognition pages 12-31, 31-45 J+M: Speech Recognition: Advanced Topics pages 11-16
10. ASR: Learning (Baum-Welch) and Disfluencies J+M: Automatic Speech Recognition 46-52 , J+M: Speech Recognition: Advanced Topics pages 1-11

Recommended Texts:

1. Jurafsky and Martin. (2014). *Speech and language processing*. (2nd ed.). Boston: Prentice Hall.

Suggested Readings

1. Taylor, Paul. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge University Press.

CSEC-7432 **Stochastic Processes** **3 (3+0)**

The goal is to teach the theoretical concepts and techniques for solving problems that arises in practice. Beginning with the random variables, this course leads to the concept of stochastic process and linear filtering of random processes. More precisely, the objectives are 1. study of the basic concepts of the theory of stochastic processes; 2. introduction of the most important types of stochastic processes; 3. study of various properties and characteristics of processes; 4. study of the methods for describing and analyzing complex stochastic models. Various practical skills acquired during the study process would be: 1. understanding the most important types of stochastic processes (Poisson, Markov, Gaussian processes and others) and ability of finding the most appropriate process for modelling in particular situations arising in economics, engineering and other fields; 2. Understanding the notions of ergodicity, stationarity, stochastic integration; application of these terms in context of financial mathematics. Course will also include few practical exercises.

Contents

1. Review of basic probability: Sample space, event, conditional probability, independent events, Baye's formula, random variable, distribution, cumulative distribution, Bernoulli distribution, binomial distribution, Poisson distribution, geometric distribution, density function, exponential distribution, gamma distribution, expectation, variance, standard deviation, joint
2. General Concepts: Definitions, Systems with Stochastic Inputs, The Power Spectrum, Discrete-Time Processes, Continuity, Differentiation, Integration, Shift Operators and Stationary Processes.
3. Finite Markov chains: basics, examples, 1-step and n-step transition probabilities, stationary distributions, classifying states, periodicity of classes, absorption of transient states, reversibility
4. Branching processes: probability generating function, compound distribution
5. Renewal theory: sequence generating function, pattern generation, consecutive successes, mean number of trials, breaking even, mean number of occurrences, comparison of patterns
6. Markov processes: Poisson processes, extensions, pure birth processes, Yule process, pure death processes, birth and death processes, linear case, linear growth with immigration, M/M/infinity queue, power supply problem, stationary distributions, examples, absorption
7. Brownian motion: definition, no drift case, reaching a point, avoiding zero, returning to zero, drift.
8. Random Walks and Other Applications: Random Walks, Cyclostationary Processes
9. The Schwarz Inequality.
10. Mean Square Estimation: Introduction, Prediction.

Pre-Requisites: Probability and Statistics

Recommended Texts

1. Bhattacharya, R. N., & Waymire, E. C. (2009). Stochastic processes with applications (vol. 61). Philadelphia: SIAM.

Suggested Readings

1. Brzezniak, Z., & Zastawniak, T. (2000). Basic stochastic processes: a course through exercises. New York: Springer Science & Business Media.
2. Nakagawa, T. (2011). Stochastic *processes: With applications to reliability theory*. New York: Springer Science & Business Media.
3. Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. New York: Tata McGraw-Hill Education.

CSEC-7433 **Stochastic Systems** **3 (3+0)**

Exploration of noise, uncertainty, and randomness in the context of signals and systems. The course will introduce discrete- and continuous-time random processes as input and/or output signals of various types of systems, with and without memory or feedback. Probabilistic notions will be tightly integrated with techniques from signals and systems, such as linearity, time-invariance, causality, transform methods, and stability. Basic concepts will be illustrated via numerous examples, such as noise in linear and nonlinear circuits, average consensus and PageRank, queuing systems, noise in remote sensing applications, Bayesian filtering, Monte Carlo simulation, risk allocation in financial portfolios, stochastic gradient descent. Upon successful completion of the course, students will be able to reason about noise, mean and variance, joint moments, sequences of random variables, stochastic convergence and limit theorems, uncertainty, and randomness in the context of engineering systems using tools and techniques from probability theory and systems theory. Student will be able to understand, analyze, and solve typical problems in statistics

Contents

1. Overview of Probability: Introduction, Definitions, Probability and Induction.
2. Causality Versus Randomness.
3. The Axioms of Probability: Set Theory, Probability Space, Conditional Probability.
4. Repeated Trials: Combined Experiments.
5. Bernoulli Trials, Bernoulli's Theorem and Games of Chance.
6. The Concept of a Random Variable: Introduction, Distribution and Density Functions, Specific Random Variables,
7. Conditional Distributions, Asymptotic Approximations for Binomial Random Variable.
8. Functions of One Random Variable: The Random Variable g(x), The Distribution of g(x).
9. Mean and Variance.
10. Moments, Characteristic Functions.
11. Two Random Variables: Bivariate Distributions, One Function of Two Random Variables, Two Functions of Two Random Variables.
12. Joint Moments, Joint Characteristic Functions, Conditional Distributions
13. Sequences of Random Variables: General Concepts.
14. Conditional Densities, Characteristic Functions, and Normality.
15. Mean Square Estimation.
16. Stochastic Convergence and Limit Theorems.
17. Random Numbers: Meaning and Generation.
18. Statistics: Introduction, Estimation.
19. Parameter Estimation, Hypothesis Testing.

Pre-Requisites: Probability and Statistics

Recommended Texts

1. Klyatskin, V. I. (2005). *Dynamics of stochastic systems*, (1st ed.). Amsterdam: Elsevier.

Suggested Readings

1. Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. New York: Tata McGraw-Hill Education.

2. Adomian, G. (Ed.). (2014). *Applied stochastic processes*. Cambridge: Academic press.

ITEC-7408 **Telecom Management** **3 (3+0)**

Telecom Management generally focuses on the basics of Telecom Management. This course helps students to acquire essential skills to manage the vast network of telecommunication. Telecom Management involves both traditional Information Technology and telecommunication lines, delineation between the two has become blurred as the two technologies have increasingly intermingled to effectively send the message. A program in telecom management includes conceptualizing innovative ideas to manage operations and marketing of telecommunications-related technologies. The purpose of this course is to provide an understanding of the key technical, managerial and policy issues in the effective development and use of telecommunication solutions by organizations. Discussion of technology and technology trends will be set in the context of applications. The interdisciplinary program offers necessary skills to manage telecom networks – voice, video and data. The course provides the understanding of the operation and management of a telecommunication business. After successful completion of the course student will be able to develop a digital logic and apply it to solve real life problems.

Contents

1. Introduction to Telecommunication
2. Management and Planning
3. Telecommunication Strategic Planning
4. Developing Requirements and Specifications
5. Managing Long-Distance Services
6. Managing Internet Services
7. Managing PBX and Key Telephone Equipment
8. Managing Automatic Call Distributors
9. Managing Voice Processing Equipment
10. Managing Local Area Network and Internets
11. Managing Wide Area Networks
12. Managing Video and Audio-Conferencing Equipment
13. Managing Convergence
14. Developing a Telecommunication Operations Plan
15. Disaster Prevention and Recovery
16. Telecommunications Project Management
17. Managing Telecommunications Security

Pre-Requisites: Computer Networks

Recommended Texts

1. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.
2. P. J. Louis. (2002). *Telecom management crash course: a telecom company survival guide*, (1st ed.). New York: McGraw-Hill Professional.

Suggested Readings

1. JayrajUgarkar. (2010). *The essentials of telecommunications management: A simple guide to understanding a complex industry*, Bloomington: AuthorHouse.

ITEC-7409 Telecommunication Systems 3 (3+0)

Telecommunications systems include wired and wireless local and wide area networks and hardware and software providing the capabilities for systems to communicate with each other or with users. The set of telecommunications systems supporting most federal government agencies includes network infrastructure and other technical solution components owned by commercial telecommunications service providers and managed on behalf of the government. Telecommunications, also known as telecom, is the exchange of information over significant distances by electronic means and refers to all types of voice, data and video transmission. This is a broad term that includes a wide range of information transmitting technologies such as telephones (wired and wireless), microwave communications, fiber optics, satellites, radio and television broadcasting, the internet and telegraphs. The modern telecommunications infrastructure—made possible by research performed over the last several decades the course provides an in-depth investigation of telecommunications system concepts and Terminology. The course covers the most popular telecommunication networks and forthcoming technologies.

Contents

1. Communication Channel and the Communication Network
2. Digital Telephony: From Analog to Digital, Advantages and Disadvantages of Digital Techniques, PCM Techniques and Standards, Low Bit-Rate Voice Coding, Voice over the Internet, Devices for Digital Telephony.
3. Switching and Signaling Systems Switching, Circuit Switching and Packet Switching, Signaling Systems, Digital Exchanges, Private Branch Exchanges and Key Phone Systems, Merging Data and Voice, Packet Switching Systems, Switching Fabric Interfaces and ICs, Optics and the Future.
4. Cellular Systems
5. Fixed Wireless Access
6. Digital Subscriber Loop
7. Transmission Techniques.
8. Telecommunication Systems Testing
9. Embedded Systems Design for Telecommunications

Pre-Requisites: Computer Networks

Recommended Texts

1. Niha Kularatna and Dileeka Dias. (2004). *Essentials of modern telecommunications systems*, London: Artech House.
2. James Harry Green (2005). The irwin handbook of telecommunications, (5th ed.). New York: McGraw-Hill Professional.

Suggested Readings

1. Lillian Goleniewski. (2002). Telecommunications Essentials: The Complete Global Source for Communications Fundamentals, Data Networking and the Internet, and Next-Generation Networks, (1st ed.). Boston: Addison-Wesley Professional.
2. Michael Fitz. (2007). *Fundamentals of Communications Systems*, (1st ed.). New York: McGraw-Hill Professional.

CSEC-7435 **Text Processing and Data Analysis** **3 (3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing, Text Processing Functions.
6. Loop Control, Hashes, DBM Databases, Advanced Sorting.
7. Regular Expressions, Environment Variables, CGI-Programming.
8. Process Management, References and Data Structures.
9. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. Sebastopol: O'Reilly & Associates, Inc.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming Perl: Unmatched power for text processing and scripting*. Sebastopol: O'Reilly Media, Inc.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl cookbook: solutions & examples for perl programmers*. Sebastopol: O'Reilly Media, Inc.
2. Lidie, S., & Walsh, N. (2002). *Mastering Perl/Tk: graphical user Interfaces in perl*. Sebastopol: O'Reilly Media, Inc.

ITEC-7410 Virtualization in Cloud Computing 3 (3+0)

The term 'Virtualization' can be used in many respects of computer. It is the process of creating a virtual environment of something which may include hardware platforms, storage devices, OS, network resources, etc. The cloud's virtualization mainly deals with the server virtualization Virtualization in Cloud Computing is making a virtual platform of server operating system and storage devices. This will help the user by providing multiple machines at the same time it also allows sharing a single physical instance of resource or an application to multiple users. This course gives an inside look as traditional physical data centers evolve and morph into virtual entities and cloud environments. Students will learn in-depth details and considerations for planning, designing, and migrating to Virtualized Data Centers (VDCs) and cloud environments. They will learn to design VDC and cloud infrastructures, maintaining the most robust and elastic computing, network, and storage environments. Course will also include few practical exercises.

Contents

1. Virtualization: Virtualization Types, Virtualization Management, Cloud Computing, Service Models, Cloud Adoption and Barriers, Return on Investment and Cloud Benefits; Typical Design Patterns and Use Cases, Design Patterns, Cloud Use Cases, Deployment Models, IaaS as a Foundation, Cloud Consumer Operating Model.
2. Cloud Management Reference Architecture: Standards, TMF eTOM, IT Infrastructure Library (ITLI), ITU-T TMN, Building Cloud Models Using Standards, Integration of Management Systems/Functions.
3. Service Fulfillment, Service Assurance, Technical Building Blocks of IaaS: IaaS Service Composition, Developing and Offering Cloud Products, Persisting Service Data.
4. Automating and Orchestration Resources: Building the Cloud, Adding Services to the Cloud, Creation and Placement Strategies, Service Life Cycle Management.
5. Cloud Capacity Management: Tetris and the Cloud, Cloud Capacity Model, Demand Forecasting, Procurement in the Cloud, Case Study - Hybrid Cloud

.Recommended Texts

1. VenkataJosyula, Malcolm Orr and Greg Page. (2011). *Cloud Computing: Automating the Virtualized Data Center*, (1st ed.). Indianapolis: Cisco Press.
2. Mickey Iqbal, MithkalSmadi, Chris Molloy, and Jim Rymarczyk. (2011). *IT virtualization best practices: A lean, green virtualized data center approach*, (1st ed.). Chicago: MC Press.

Suggested Readings

1. Aidan Finn, Hans Vredevoort, Patrick Lownds, and Damian Flynn. (2012). *Microsoft private cloud computing*, (1st ed.). Hoboken: Sybex.
2. David Marshall, Wade A. Reynolds, and Dave McCrory. (2006). *Advanced server virtualization: VMware and microsoft platforms in the virtual data cente*r, (1st ed.). Boca Raton: Auerbach Publications.

ITCC-7402 **Information Privacy and Security** **3 (3+0)**

Information security and privacy issues are increasingly central parts of the information policy landscape. Privacy is a broad topic that covers many disciplines, stakeholders, and concerns. This course addresses the intersection of privacy and information technology, surveying a wide array of topics of concern for research and practice in the information fields. A strong need is arising for individuals with the training and skills to work in this unsettled and evolving environment. Students will learn basic concepts of "information security" along with means of protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide Integrity, confidentiality. The students will learn a balanced mix of technical and managerial issues makes this course appealing to attendees who need to understand the salient facets of information security basics and the basics of risk management. We will touch on relationships between privacy, security, and risk. Course will also include few practical exercises.

Contents:

1. Overview of e-security: Threats, risks, consequences, Sources of threats.
2. Attacks classification, Preventive measures, remedial measures.
3. Cryptography for e-security: Historical perspective.
4. Confusion vs. diffusion.
5. Stream ciphers vs. block ciphers.
6. Keys and key management, Key exchange (peer to peer, peer - key server - peer), DiffieHelman key sharing scheme.
7. Symmetric key cryptography vs asymmetric key cryptography, Trapdoor functions,
8. GPG: Overview of GPG, Commands and CLI, GPG trust model, GUI – KGPG, Seahorse, Frontends – Kleopatra, enigmail.
9. Practical applications: PKI, CA. X509 certificates, SSL/TLS, HTTPS, IPV6 and IPSEC, Proxies and Firewalls, Misc. techniques:
10. Encryption using non-cryptographic tools (vi, zip).
11. Authentication principles and methods, Passwords, two-factor authentication, One-way encryption.
12. Steganography, Hamming, Chaffing and Winnowing.
13. Management aspects: System Administration policies.
14. Security audit.
15. Penetration testing and ethical hacking.
16. Mandatory Access control, Discretionary Access Control.

Recommended Texts

1. William Stallings. (2005). *Cryptography and network security,* (4^{th} ed.). New York: Pearson Education.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. (1996). *Handbook of applied cryptography.* Boca Raton: CRC Press.

Suggested Readings

1. William Stallings. (2014). *Cryptography and network security: principles and practice,* (6^{th} ed.). New

York: Pearson Education.

ITCC-7403 **Cryptography** **3 (3+0)**

Cryptography constitutes today a fundamental and ingrained part of the security of all modern communication. Everything from web browsing, email, and telephony, to messaging apps, data storage and video conferencing, is today secured by cryptographic techniques. The modern study of cryptography investigates techniques for facilitating interactions between distrustful entities. In this course we introduce some of the fundamental concepts of this study. Emphasis will be placed on the foundations of cryptography and in particular on precise definitions and proof techniques. This will be a theory course. The learner will gain knowledge about securing both clean and corrupted systems, protect personal data, and secure computer networks. The learner will develop an understanding of security policies (such as confidentiality, integrity, and availability), as well as protocols to implement such policies. Students will be able to provide security of the data over the network, can do research in the emerging areas of cryptography and network security, implement various networking protocols and protect any network from the threats in the world.

Contents:

1. Elementary number theory: Prime numbers, Factoring.
2. Modular arithmetic, Fermat's & Euler's theorems, gcd.
3. Euclid's algorithm, Discrete logarithm problem.
4. Public key encryption:
5. Public key crypto systems, RSA algorithm.
6. Elliptic Curve cryptography, Hash digests: Properties of cryptographic hash functions.
7. MerkleDamgard construction, md family, sha family, Digital signatures, sha3, Block ciphers:
8. Block cipher principles.
9. Feistel networks, S boxes and P boxes.
10. Block cipher modes of operation, DES, 3DES, AES.
11. Message Integrity.
12. Interactive Proofs, Zero-Knowledge Proofs, Zero-Knowledge Proofs of Knowledge,
13. Non-Interactive Zero-Knowledge Proofs.
14. Secure Protocols, Two-Party Secure Computation.
15. Multiparty Secure Computation,
16. Chosen Ciphertext Security

Recommended Texts

1. William Stallings. (2005). *Cryptography and network security*, (4th ed.). New York: Pearson Education.

Suggested Readings

1. Margaret Cozzens, Steven J Miller. (2013). *The mathematics of encryption*. Providence: American Mathematical Society.

2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. (1996). *Handbook of applied cryptography*. Boca Raton: CRC Press.
3. William Stallings. (2014). *Cryptography and network security: principles and practice*, (6th ed.). New York: Pearson Education.

ITCC-7404 **Advanced Database Management Systems** **3 (3+0)**

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Database systems are important to business, industry and science. IT professionals need to critically assess the opportunities and implications presented by such systems. Effective collection, analysis, and maintenance of data is key to achieve rapid progress in almost all disciplines of science and engineering. In this course we will cover the core principles and techniques of data and information management. Make Students fully understand the concepts and technical issues of Database Administration, Management System and can Administrator Huge Database have implemented in a DBMS. Students will be able to explain and evaluate the fundamental theories and requirements that influence the design of modern database systems, assess and apply database functions and packages suitable for enterprise database development and database management, critically evaluate alternative designs and architectures for databases and data warehouses, discuss and evaluate methods of storing, managing and interrogating complex data, explain and critically evaluate database solutions for data exchange.

Contents

1. Overview of Database Systems: History, Advantages of a DBMS, Levels of Abstraction, Queries in a DBMS, Transaction Management, Structure of a DBMS.
2. The Relational Model: Introduction to the Relational Model, Integrity Constraints over Relations, Integrity Constraints, Querying Relational Data, ER to Relational, Introduction to Views.
3. Relational Algebra: Introduction to Relational Algebra, Introduction to Relational Calculus.
4. Evaluating Relational Operators: The Selection Operation, General Selection Conditions
5. Relational Query Optimizer: Translating SQL Queries into Algebra, Estimating the Cost of a Plan, Relational Algebra Equivalences, Enumeration of Alternative Plans, Nested Subqueries
6. Overview of Transaction Management: Concurrency Control, Crash Recovery.
7. Parallel and Distributed Databases: Architectures, Parallel Query Evaluation
8. Object-Database Systems: Introduction, ORDBMS Implementation Challenges
9. Deductive Databases: Introduction to Recursive Queries, Data log, Theoretical Foundations
10. Data Warehousing and Decision Support: Introduction, OLAP.
11. Data Mining: Introduction, Counting Co-occurrences, Mining for Rules, Tree-Structured Rules, Clustering, Similarity Search over Sequences, Incremental Mining and Data Streams.
12. Information Retrieval and Xml Data: Colliding Worlds.
13. Spatial Data Management, Advanced Transaction Processing, Mobile Databases.

Recommended Texts

1. Raghu Ramakrishnan, Johannes Gehrke. (2002) *Database management systems*, (3rd ed.). New York: McGraw-Hill.

Suggested Readings

1. Priscilla Walmsley. (2007) *XQuery*. Sebastopol: O'Reilly Media, Inc.
2. Rigaux, Philippe, M. Scholl, Voisard. (2001). *Spatial databases: with application to GIS*. Amsterdam:

Elsevier.

SECC-7406 **Advanced Requirements Engineering** **3 (3+0)**

Requirements engineering (RE) refers to the process of defining, documenting and maintaining requirements to the sub-fields of systems engineering, software engineering concerned with this process. Despite the fundamental principles of requirements engineering being relatively well-defined, ‘poor requirements’ are regularly held up as the reason for project failures. Experienced business analysts know that this can be due to many factors, including failure to align requirements with business objectives, address cultural issues and document requirements at the correct level. Students will be able to understanding the enterprise, the analysis portfolio, the requirements engineering plan, The requirements taxonomy, non-functional requirements. How to write requirements, check their completeness, using models to check completeness, SRS document, Vision and Scope document, Requirement specification in agile projects, Eliciting data requirements, Modeling/specifying data requirements, Business Data Diagram, Data Dictionary, Data Flow Diagram, Software quality attributes, defining exploring and specifying software quality attributes, their relationship to functional requirements, Requirements review techniques, validating requirements against acceptance criteria.

Contents

1. Software Requirements Fundamentals: Product and process requirements, Functional and non-functional requirements, Emergent properties, Quantifiable requirements, System and software requirements.
2. Requirements Process: Process models, Process actors, Process support and management, Process quality and improvement. Requirements Analysis: Requirements sources, Elicitation techniques.
3. Requirements Analysis: Requirements classification, Conceptual modeling, Architectural design and requirements allocation, Requirements negotiation, Formal analysis.
4. Requirements Specification: System definition document, System requirements document, Software requirements specification. Requirements Validation: Requirements reviews, Prototyping, Model validation, Acceptance tests.
5. Practical Considerations: Iterative nature of the requirements process, change management, Requirements attributes, Requirements tracing, Measuring requirements. Software Requirements Tools.
6. Current research topics in requirement engineering.

Recommended Texts

1. Roger S. Pressman, Bruce R. Maxim. (2015). *Software engineering: a practitioner’s approach*, (8th ed.). New York: McGraw-Hill Education.

Suggested Readings

1. Hassan Gomaa. (2011). *Software modeling and design: UML, use cases, patterns, and software architectures,* Cambridge: Cambridge University Press.
2. Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates. (2004). *Head first design patterns.* Sebastopol: O’Reilly Media, Inc.

SECC-7407 Advanced Software System Architecture 3 (3+0)

Student will be able to describe the importance of predictive architecting early in the system life-cycle state/describe the applicability of model-based approaches - describe the intentionality of models throughout the system life-cycle - describe/state the characteristics and challenges of architecting system-of-systems and ultra-large-scale systems - distinguish between software architecture, enterprise architecture, system architecture, and run-time architectures. - explain the link between business strategy, business process and system- and software architecture - know of available tools for supporting architecture modelling and analysis - explain the role of architecture as a central artefact in system development with the key objectives are to develop understanding about the fundamental architectural concepts in context of software and system and to enable students to effectively evaluate the architectures of various kinds of systems. Particular attention will be devoted to important topics, such as architectural styles and patterns, domain specific software architectures, modelling techniques, visualization and implementation of architectures, and networked and distributed architectures.

Contents

1. Quality attributes in the context of architecting.
2. Qualitative and quantitative assessment of architectures.
3. Architectural modeling through Architecture Description Languages.
4. System modeling its relation to software architecting.
5. Architecting for evolution and variability.
6. Partitioned and layered architectures.
7. System-of-Systems
8. Ultra-Large-Scale Systems.
9. Software Product Lines and Configurable Software.
10. Self-Adaptive Software.
11. Architectural Description Languages.
12. Feature Modeling. Architecture
13. Model-Based Testing.
14. Current research topics in software system architecture.

Recommended Texts

1. Humberto Cervantes, Rick Kazman (2016). *Designing software architectures: A practical approach,* (1st ed.). Boston: Addison-Wesley Professional.
2. P. Clements and L. Northrup (2002). *Software product lines: practices and patterns.* Boston: Addison-Wesley.

Suggested Readings

1. R. Taylor, N. Medvidović and E.M. Dashofy (2010). *Software architecture: foundations, theory, and practice.* Hoboken: John Wiley.

SECC-7408 Software Testing and Quality Assurance 3 (3+0)

The only way to minimize the risk of defects while also maximizing end-user experience is by including software and quality assurance testing throughout the entire development process. Products must be tested in different ways, with different users and different scenarios to make sure that the software that end-users receive is a consistent, high-quality experience in a range of situations. The course was designed to bring focus to QA and testing since many jobs and opportunities are available in this area. The course covers both technical foundations and tools, as well as managerial and organizational aspects. More specifically, it answers questions as: What is a test case?, What is a test suite?, What is a software testing?, What is the difference between software testing and quality assurance?, What is positive and negative testing?, What is the difference between white box and black box testing?, What are the different types of testing?

Contents

1. Testing techniques.
2. Black Box testing,
3. White Box
4. Grey Box testing techniques.
5. Software system quality components and activities that support software quality
6. QA objectives: reliability, correctness, testability, maintainability, flexibility, portability, efficiency, integrity, usability, reusability, and interoperability.
7. Theoretical background: program correctness proofs, cyclomatic complexity, software reliability modelling.
8. Software unit testing to verify unit specifications
9. Integration testing to verify design specifications
10. System testing to verify requirements specifications
11. Usability testing
12. Performance, reliability, and regression testing
13. Alpha, beta and acceptance testing
14. Software testing and quality assurance tools.
15. Open source testing using NUnit and JUnit.
16. Quality Assurance planning and execution.
17. Automated testing topics include constructing a framework, scripting techniques, generating a test data, generating test architecture, pre/post-processing, test maintenance, and job specific metrics.
18. Current research topics in Software Testing and Quality Assurance.

Recommended Texts

1. Mark Utting and Bruno Legeard. (2006). *Practical model-based testing: A tools approach*. San Francisco: Morgan Kaufmann Publishers.

Suggested Readings

1. Jeff Tian. (2005). *Software quality engineering, testing, quality assurance, and quantifiable improvements*. Piscataway: IEEE Computer Society.
2. P Ammann and J Offutt. (2008). *Introduction to software engineering*. Cambridge: Cambridge University Press.

SECC-7409 Advanced Software Project Management 3 (3+0)

To develop students' ability to plan and manage software development projects successfully, maximizing the return from each stage of the software development life cycle. This course introduces software engineers aspiring to become technical team leaders or software project managers to the responsibilities of these roles. Learn about the different types of software projects that exists and learn about the challenges that each of these types of projects poses in front of a project manager. Get to know the different and current methodologies used in the business space and know about your roles and responsibilities as a software project manager. After completion of course students will be able to learn successful software development management, including organizing the software development team; interfacing with other engineering organizations (systems engineering, quality assurance, configuration management, and test engineering); assessing development standards; selecting the best approach and tailoring the process model; estimating software cost and schedule; planning and documenting the plan; staffing the effort; managing software cost and schedule during development; risk engineering; and continuous process improvement.

Contents

1. Comparison of PM Tools.
2. PMI's Knowledge Areas
3. PMI Framework
4. PMI Process Groups.
5. Project Planning
6. Project Evaluation
7. Building Teams
8. Tracking and control
9. Comparison of Software Effort Estimation techniques
10. Comparison of different software development and designing models.
11. Analysing different risk management methodologies.
12. Studying latest conference and journal papers related to state-of-the-art techniques.

Recommended Texts

1. Bob Hughes and Mike Cotterell (2009) *Software project management*, (5^{th} ed.). New York: McGraw-Hill Education.
2. Jeff Tian. (2005). *Software quality engineering, testing, quality assurance, and quantifiable improvements*. Piscataway: IEEE Computer Society.

Suggested Readings

1. Robert K. Wysocki (2011) *Effective project management: traditional, agile, extreme*, (6^{th} ed.). Hoboken: Wiley.
2. Dwayne Phillips (2004) *The software project manager's handbook - Principles that work at work*,

(2nd ed.). Piscataway: IEEE Computer Society Press and Wiley Inter-science.

CSEC-7436 Swarm Intelligence 3(3+0)

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. The course is aimed to provide students' knowledge of swam principles, experiments and models. They will learn about mobile robotics, machine-learning techniques, Multi-level modeling of self-organized robotic systems, combined modeling and machine-learning methods for control optimization, aggregation and segregation and Distributed structure building in natural and artificial systems. More specifically students would study to evaluate the power and limitation of Swarm Intelligence when it comes to solving real life problems, to advance the state of the art in Swarm Intelligence, and to introduce students to academic research. There will be a course project in which teams of students apply swarm intelligence to a problem of their choosing. This application should advance the state of knowledge in swarm intelligence. You will be asked to research, design and build a swarm system. Course will also include few practical exercises.

Contents

1. Introduction to swarm intelligence and key principles (e.g., self-organization, stigmergy), natural and artificial examples, computational and embedded SI. Foraging, trail laying/following mechanisms. Open-space, multi-source foraging experiments: biological data and microscopic models.
2. From real to virtual ants
3. Introduction to mobile robotics
4. Introduction to unsupervised multi-agent machine-learning techniques for automatic design and optimization: terminology and classification, Genetic Algorithms (GA) and Particle Swarm optimization (PSO). Comparison between both techniques in theory and practice.
5. Application of machine-learning techniques to automatic control design and optimization in single-robot and multi-robot experiments. Specific issues for automatic control design and optimization in collective systems
6. Collective movements in natural societies; focus on flocking phenomena. Collective movements in artificial systems: Reynolds' virtual agents (Boids) and experiments with multi-robot systems
7. Multi-level modeling of self-organized robotic systems: microscopic and macroscopic models; Markov formalism; linear and nonlinear micro-to-macro mapping, model analysis.
8. Combined modeling and machine-learning methods for control optimization. Diversity and specialization metrics.
9. Division of labor and task-allocation mechanisms, part I: threshold-based algorithms.

Suggested Readings

3. Robert K. Wysocki (2011) *Effective project management: traditional, agile, extreme*, (6th ed.). Hoboken: Wiley.
4. Dwayne Phillips (2004) *The software project manager's handbook - Principles that work at work*, (2nd ed.). Piscataway: IEEE Computer Society Press and Wiley Inter-science.

Recommended Texts

1. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz. (1999). *Swarm intelligence: from natural to artificial systems*, Oxford: Oxford University Press.

CSEC-7437 **Evolutionary Computation** **3(3+0)**

It is intended as a fundamental introduction, as well as a survey of the many aspects of evolutionary algorithms (EAs), in particular GA, GP, ES, and will concentrate on the basic concepts of representation, operators and overall control, followed by examples of the use of these concepts in important applications. As such, this course will not do much in-depth study of any one area. Rather the course is intended as a good introduction for those who have had no exposure to EC and as a stepping stone for those interested in more specific areas. At the end of this course, student would be able to: Explain evolutionary computation techniques and methodologies set in the context of modern heuristic methods. , apply various evolutionary computation methods and algorithms for particular classes of problems, develop evolutionary algorithms for real-world applications, and use scientific research papers and present them in a seminar talk.

Contents

1. Introduction to Evolutionary Computation, Biological and artificial evolution, Evolutionary computation and AI, Different historical branches of EC, e.g., GAs, EP, ES, GP, A simple evolutionary algorithm.
2. Search Operators, Recombination/Crossover for strings (e.g., binary strings), e.g., one-point, multi-point, and uniform crossover operators, Mutation for strings, e.g., bit-flipping, Recombination/Crossover and mutation rates, Recombination for real-valued representations, e.g., discrete and intermediate recombination, Mutation for real-valued representations, e.g., Gaussian and Cauchy mutations, self-adaptive mutations
3. Selection Schemes, Fitness proportional selection and fitness scaling, Ranking, including linear, power, exponential and other ranking methods, Tournament selection
4. Search Operators and Representations, Mixing different search operators, An anomaly of self-adaptive mutations, The importance of representation, e.g., binary vs. Gray coding
5. Evolutionary Combinatorial Optimization, Evolutionary algorithms for TSPs, Evolutionary algorithms for lecture room assignment, Hybrid evolutionary and local search algorithms
6. Co-evolution, Cooperative co-evolution, Competitive co-evolution
7. Niching and Speciation, Fitness sharing (explicit and implicit), Crowding and mating restriction
8. Constraint Handling, Common techniques, e.g., penalty methods, repair methods
9. Genetic Programming, Trees as individuals, Major steps of genetic programming, e.g., functional and terminal sets, initialization, crossover, mutation, fitness evaluation, etc. Search operators on trees, Automatically defined functions, Issues in genetic programming
10. Multi objective Evolutionary Optimization, Pareto optimality

Recommended Texts

1. Andries Engelbrecht (2007) *Computational intelligence: An introduction.* Hoboken: Wiley.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings

1. Z. Michalewicz and D. B. Fogel, (2000) *How to solve it: Modern heuristics,* New York: Springer.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

CSEC-7438 **Fog Computing** **3(3+0)**

Fog computing has emerged because cloud computing hasn't been able to keep up with the growth of IoT devices. The increase of consumer and enterprise devices connected to the IoT has put too much strain on cloud services from even the most cutting-edge providers. The reason is that the cloud is too far away from the point of origin, and sending data to the cloud to be analyzed results in latency that is simply unacceptable in many environments. This course is designed to give a multitude of technologies that comprise the modern concept of Fog and Edge computing, their applications, architectures, and technologies. Fog Computing is an emergent architecture for computing, storage, control, and networking that distributes these services closer to end users along the cloud-to-things continuum. It covers both mobile and wire line scenarios, traverses across hardware and software, resides on network edge but also over access networks and among end users, and includes both data plane and control plane.

Contents

1. Overview: From Cloud to Fog, Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers
2. Overview: From IT to loT to IoE, Basic attributes and challenges for IoE
3. Fog/Edge Computing ComputingArchitectures, Feasible Service Models of Fog
4. Fog/Edge Computing Applications: Smart city, Smart health, Smart Building, Internet of Energy
5. Fog/Edge/IoT Data storage and Management, Real-time streaming data processing
6. Resource-constrained devices management and optimization
7. Scheduling for Fog/Edge infrastructures
8. Modeling and Simulation of Fog and Edge Computing Environments using iFogSim Toolkit:
9. Components of iFogsim: physical components and logical components
10. Installation of iFogSim, Building Simulation with iFogSim
11. Application Case Studies: a latency-sensitive online game, Intelligent Surveillance through Distributed Camera Networks, Performance Evaluation
12. Example Scenarios: Sensors with different tuple emission rate, send specific number of tuples from a sensor, connect lower level Fog devices with nearby gateways, Mobility of a Fog device, Make Cluster of Fog devices, Simulation of a Placement Policy

Recommended Texts

1. Rajkumar Buyya, Satish Srirama. (January, 2019). *Fog and edge computing: principles and paradigms*, Hoboken: Wiley Press.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings

1. Z. Michalewicz and D. B. Fogel, (2000) *How to solve it: Modern heuristics*, New York: Springer.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

CSEC-7439 Advance Natural Language Engineering 3 (3+0)

In this course we will explore statistical, model-based approaches to natural language processing. There will be a focus on corpus-driven methods that make use of supervised and unsupervised machine learning approaches and algorithms. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. In this course, students will gain a basic introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Text Classification via Naive Bayes & Maximum Entropy, Classification Tutorial, MaxEnt Tutorial, Generative and Discriminative Classifiers.
2. Hidden Markov Models, Clustering, Part-of-Speech Tagging, TnT Tagger.
3. Advanced Part-of-Speech Tagging, Word Alignments, MT Tutorial, Overview, IBM Models, HMM-Alignments, Agreement.
4. Phrase-Based Translation, Decoding, Phrases
5. Syntactic Parsing, Best-First, A*, K-Best, Shift-Reduce Parsing.
6. Advanced Constituency Parsing, Unlexicalized, Lexicalized, Latent Variable.
7. Semantic Parsing, Hierarchical (Syntax-Based) Translation, Hiero, GHKM, Syntax vs. Phrases, Synchronous Grammars.
8. Sentiment Analysis, Aspects, Lexicons, Summarization.
9. Summarization, Query, N-Gram, Topical.
10. Lexical Acquisition.

Recommended Texts

1. Jurafsky and Martin. (2014). Speech and language processing. (2nd ed.). Boston: Prentice Hall.
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. Cambridge: MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. Sebastopol: OReilly Media.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge: MIT press.

CSEC-7440 **Special Topics in Natural Language Engineering** **3 (3+0)**

This course covers a broad range of topics in computational linguistics/natural language processing, including word and sentence tokenization, text classification and sentiment analysis, spelling correction, information extraction, parsing, meaning extraction, and question answering, We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. In recent years, deep learning (or neural network) approaches have obtained very high performance across many different NLP tasks, using single end-to-end neural models that do not require traditional, task-specific feature engineering. In this course, students will gain a thorough introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Logistic Regression: Classification: The sigmoid; Learning in Logistics Regression.
2. The Cross-Entropy loss function; Gradient Descent, Regularization, etc.
3. Vector Semantics: Lexical Semantics, Vector Semantics, Words and Vectors.
4. Cosine for measuring similarity, TF-IDF Weighting terms in the vector, etc.
5. Neural Networks and Neural Language Models.
6. Units, The XOR problem, Feed Forward Neural Network, Training Neural Nets, etc.
7. Part-of-Speech Tagging.
8. Word Classes, Penn POS Tagset, POS Tagging, HMM POS Tagging, etc.
9. Sequence Processing with Recurrent Networks Simple recurrent neural network, Applications of Recurrent Neural Networks.
10. Deep Networks: Stacked and Bidirectional RNNs, Managing Context in RNNs: LSTMs and GRUs, Words, Subwords and Characters.
11. Encoder-Decoder Models.
12. Attention, and Contextual Embeddings

Recommended Texts

1. Jurafsky and Martin. (2014). Speech and language processing. (2nd ed.). Boston: Prentice Hall.
2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. Cambridge: MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. Sebastopol: OReilly Media.
2. Koehn, P. (2009). Statistical machine translation. Cambridge: Cambridge University Press.
3. Jelinek, F. (1997). Statistical methods for speech recognition. Cambridge: MIT press.

CSEC-7409 **Computational Intelligence** **3(3+0)**

The course is aimed to provide students' knowledge of up-to-date efficient variants of different meta heuristics together with their potential applications. Also, to develop the interest of students in both research and development. The course goal is to make students familiar with basic principles of various computational methods of data processing that can commonly be called computational intelligence (CI). Here belong mainly bottom-up approaches to solutions of (hard) problems based on various heuristics (the so-called soft computing), rather than exact approaches of traditional artificial intelligence based on logic (hard computing). Examples of CI are nature-inspired methods (neural nets, evolutionary algorithms), fuzzy systems, as well as various probabilistic methods under uncertainty (e.g. Bayesian models) and machine learning methods (e.g. reinforcement learning). After the course the students will be able to conceptually understand the important terms and algorithms of CI, such that they would be able to choose appropriate method(s) for a given task.

Contents

1. Introduction to Computational Intelligence topics, Fundamental concepts
2. Introduction to Evolutionary Computing, Evolutionary computation terms, Canonical genetic algorithm
3. Evolutionary computation variations, Evolutionary programming and Evolution strategies
4. Implementation of Evolutionary Computing
5. Swarm Intelligence, Particle Swarm Optimization
6. Classification, Learning, and Adaptation, Supervised, Unsupervised, Reinforcement Learning
7. Data partitioning and Cross Validation, Error metrics: Mean squared error, receiver operating characteristic curves, Neural Networks and Evolutionary Computation: Explanation and Sensitivity Analysis
8. Neural Networks Implementation, Evolutionary and swarm-based neural networks
9. Fuzzy sets, Membership functions, linguistic variables
10. Fuzzy Logic, Fuzzy set operators, Fuzzy rule-based systems
11. Fuzzification, defuzzification, Fuzzy control, Evolving fuzzy rule systems
12. Basic ANN Code, Neuro-fuzzy systems
13. Fuzzy-GA systems, Support Vector Machines
14. Probabilistic reasoning: Bayesian reasoning and Dempster-Shafer theory
15. Bayesian belief networks, Fuzzy belief networks, Evolving belief networks
16. Artificial Immune Systems

.

Recommended Texts

1. Russell Eberhart and Yuhui Shi. (2007). *Computational intelligence: concepts to implementations*. Amsterdam: Elsevier.
2. Fakhreddine Karray and Clarence de Silva. (2004). *Soft computing and intelligent systems design*. New York: Pearson Edition.

Suggested Readings

1. Amit Konar. (2005). *Computational intelligence: principles, techniques, and applications.* New York: Springer Science & Business Media.

MS

INFORMATION TECHNOLOGY

ITCC-7401 Advanced Topics in Wireless Networks 3(3+0)

This course will cover the fundamental aspects of wireless networks, with emphasis on current and next-generation wireless networks. This will cover the latest research in the area of wireless networking. These types of networks have been growing exponentially in the past several years. The advent of wireless and wired networks convergence and as the Internet is increasingly becoming the tool for a wide range of technical, economical and industrial applications, resources management becomes very crucial and vital issue for any future networks. The objectives of this course are to focus on resource management and performance analysis in transporting multimedia traffic in wireless communication networks. Topics include: traffic characteristics, connection admission control, packet scheduling, access control, and mobility and handoff management. The class will build understanding of all layers of wireless networking and the interactions between them. We will introduce the students to wireless networking research and guide them to investigate novel ideas in the area via semester-long research projects.

Contents

1. Introduction to Radio Resource Management for Wireless Networks
2. Resource Management for Circuit-Switched Services
3. Traffic Modeling
4. Access Control and Admission Control
5. Mobility Management and Handoff Management
6. MAC and Packet Transmission Scheduling
7. Resource Management in Ad Hoc Networks
8. Applications: Resource Management in Packet Access
9. CDMA2000 & WCDMA (i.e., UMTS) Systems
10. Qos and QoE
11. Internet of Things (IOT)
12. WRAN
13. CRAN,
14. Vehicular Adhoc Network

Recommended Texts:

1. J. Zander and S.-L. Kim, Radio. (2001). *Resource management for wireless networks*, Norwood: Artech House Publishers.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings:

1. T. Janevski. (2003). *Traffic analysis and design of wireless IP networks*. Jamalipour, The Wireless Mobile Internet, Hoboken: Wiley.

2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

ITCC-7402 Information Privacy and Security 3(3+0)

Information security and privacy issues are increasingly central parts of the information policy landscape. Privacy is a broad topic that covers many disciplines, stakeholders, and concerns. This course addresses the intersection of privacy and information technology, surveying a wide array of topics of concern for research and practice in the information fields. A strong need is arising for individuals with the training and skills to work in this unsettled and evolving environment. Students will learn basic concepts of "information security" along with means of protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide Integrity, confidentiality. The students will learn a balanced mix of technical and managerial issues makes this course appealing to attendees who need to understand the salient facets of information security basics and the basics of risk management. We will touch on relationships between privacy, security, and risk.

Contents:

1. Overview of e-security: Threats, risks, consequences, Sources of threats.
2. Attacks classification, Preventive measures, remedial measures.
3. Cryptography for e-security: Historical perspective.
4. Confusion vs. diffusion.
5. Stream ciphers vs. block ciphers.
6. Keys and key management, Key exchange (peer to peer, peer - key server - peer), DiffieHelman key sharing scheme.
7. Symmetric key cryptography vs asymmetric key cryptography, Trapdoor functions,
8. GPG: Overview of GPG, Commands and CLI, GPG trust model, GUI – KGPG, Seahorse, Frontends – Kleopatra, enigmail.
9. Practical applications: PKI, CA. X509 certificates, SSL/TLS, HTTPS, IPV6 and IPSEC
10. Encryption using non-cryptographic tools (vi, zip).
11. Authentication principles and methods, Passwords, two-factor authentication, One-way encryption.
12. Steganography, Hamming, Chaffing and Winnowing.
13. Management aspects: System Administration policiesSecurity audit.
14. Penetration testing and ethical hacking, Mandatory Access control, Discretionary Access Control.
15. Monitoring and logging tools.

Recommended Texts

1. William Stallings. (2005). *Cryptography and network security*, (4th ed.). New York: Pearson Education.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. (1996). *Handbook of applied cryptography.* Boca Raton: CRC Press.

Suggested Readings

1. Margaret Cozzens, Steven J Miller. (2013). *The mathematics of encryption*. Providence: American Mathematical Society.
2. William Stallings. (2014). *Cryptography and network security: principles and practice*, (6th ed.). New York: Pearson Education.

ITCC-7403 **Cryptography** **3(3+0)**

Cryptography constitutes today a fundamental and ingrained part of the security of all modern communication. Everything from web browsing, email, and telephony, to messaging apps, data storage and video conferencing, is today secured by cryptographic techniques. The modern study of cryptography investigates techniques for facilitating interactions between distrustful entities. In this course we introduce some of the fundamental concepts of this study. Emphasis will be placed on the foundations of cryptography and in particular on precise definitions and proof techniques. This will be a theory course. The learner will gain knowledge about securing both clean and corrupted systems, protect personal data, and secure computer networks. The learner will develop an understanding of security policies (such as confidentiality, integrity, and availability), as well as protocols to implement such policies. Students will be able to provide security of the data over the network, can do research in the emerging areas of cryptography and network security, implement various networking protocols and protect any network from the threats in the world.

Contents:

1. Elementary number theory: Prime numbers, Factoring.
2. Modular arithmetic, Fermat's & Euler's theorems, gcd.
3. Euclid's algorithm, Discrete logarithm problem.
4. Public key encryption:
5. Public key crypto systems, RSA algorithm.
6. Elliptic Curve cryptography, Hash digests: Properties of cryptographic hash functions.
7. MerkleDamgard construction, md family, sha family, Digital signatures, sha3, Block ciphers:
8. Block cipher principles.
9. Feistel networks, S boxes and P boxes.
10. Block cipher modes of operation, DES, 3DES, AES.
11. Message Integrity, Interactive Proofs, Zero-Knowledge Proofs
12. Non-Interactive Zero-Knowledge Proofs.
13. Secure Protocols, Two-Party Secure Computation.
14. Multiparty Secure Computation,
15. Chosen Ciphertext Security

Recommended Texts

1. William Stallings. (2005). *Cryptography and network security*, (4^{th} ed.). New York: Pearson Education.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings

1. Margaret Cozzens, Steven J Miller. (2013). *The mathematics of encryption*. Providence: American Mathematical Society.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. (1996). *Handbook of applied cryptography.* Boca Raton: CRC Press.
3. William Stallings. (2014). *Cryptography and network security: principles and practice*, (6^{th} ed.). New York: Pearson Education.

ITCC-7404 **Advanced Database Management Systems** **3(3+0)**

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Database systems are important to business, industry and science. IT professionals need to critically assess the opportunities and implications presented by such systems. Effective collection, analysis, and maintenance of data is key to achieve rapid progress in almost all disciplines of science and engineering. In this course we will cover the core principles and techniques of data and information management. Make Students fully understand the concepts and technical issues of Database Administration, Management System and can Administrator Huge Database have implemented in a DBMS. Students will be able to explain and evaluate the fundamental theories and requirements that influence the design of modern database systems, assess and apply database functions and packages suitable for enterprise database development and database management, critically evaluate alternative designs and architectures for databases and data warehouses, discuss and evaluate methods of storing, managing and interrogating complex data, explain and critically evaluate database solutions for data exchange.

Contents

1. Overview of Database Systems: History, Advantages of a DBMS, Levels of Abstraction
2. The Relational Model: Introduction to the Relational Model
3. Relational Algebra: Introduction to Relational Algebra, Introduction to Relational Calculus.
4. Evaluating Relational Operators: The Selection Operation, General Selection Conditions
5. Relational Query Optimizer: Translating SQL Queries into Algebra, Estimating the Cost of a Plan
6. Overview of Transaction Management: Concurrency Control, Crash Recovery.
7. Parallel and Distributed Databases: Architectures, Parallel Query Evaluation, Parallelizing Individual Operations, Parallel Query Optimization, Distributed Databases.
8. Object-Database Systems: Introduction, ORDBMS Implementation Challenges
9. Deductive Databases: Introduction to Recursive Queries, Datalog, Theoretical Foundations, Recursive Queries with Negation, From Datalog to SQL, Evaluating Recursive Queries.
10. Data Warehousing and Decision Support: Introduction, OLAP.
11. Data Mining: Introduction, Counting Co-occurrences, Mining for Rules, Tree-Structured Rules, Clustering, Similarity Search over Sequences, Incremental Mining and Data Streams.
12. Information Retrieval and Xml Data: Colliding Worlds.
13. Spatial Data Management, Advanced Transaction Processing, Mobile Databases.

Recommended Texts

1. Raghu Ramakrishnan, Johannes Gehrke. (2002) *Database management systems*, (3rd ed.). New York: McGraw-Hill.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings

1. Priscilla Walmsley. (2007) *XQuery*. Sebastopol: O'Reilly Media, Inc.

2. Rigaux, Philippe, M. Scholl, Voisard. (2001). *Spatial databases: with application to GIS*. Amsterdam: Elsevier.

ITEC-7401 Business Process Management Trends & Technologies 3 (3+0)

Business process and business process management concepts have matured over the years and new technology, concepts, standards, and solutions appear. The inclusion of big data and analytics have brought about enormous changes in the way's organizations operate and provide training to their employees. The deployment of BPM software generates a massive amount of data, which can be used for process optimization and performance improvement. BPM analytics is widely used for decision making, optimization of internal operations, and monitoring the performance of the process. The course will cover topics fundamentals and principles of Business Process Management. The course is designed to achieve following objectives: Understand the key terms and concepts in BPM, Learn the major methodologies and techniques for implementing BPM, Discover the various technologies that support BPM, Learn what a BPM management and process-centric organization is and how it, works, Understand the metrics and measurements critical to managing processes, Learn how to identify critical processes

Contents

1. Introduction: Motivation and Definitions, Business Process Lifecycle, Classification of Business Processes, Goals, Structure, and Organization.
2. Evolution of Enterprise Systems Architectures: Traditional Application Development, Enterprise Applications and their Integration, Enterprise Modeling and Process Orientation
3. Business Process Modeling: Foundation, Conceptual Model and Terminology, Abstraction Concepts, From Business Functions to Business Processes, Activity Models and Activity Instances
4. Process Orchestrations: Control Flow Patterns, Petri Nets, Event-driven Process Chains, Workflow Nets, Graph-Based Workflow Language, Business Process Model and Notation.
5. Process Choreographies: Motivation and Terminology, Development Phases, Process Choreography Design, Process Choreography Implementation, Service Interaction Patterns
6. Properties of Business Processes: Data Dependencies, Object Lifecycle Conformance, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness.
7. Business Process Management Architectures: Workflow Management Architectures, Flexible Workflow Management.

Recommended Texts

1. Mathias Weske. (2012). *Business process management: concepts, languages, architectures*, (2nd ed.). New York: Springer.
2. Yvonne Lederer Antonucci. (2009). *Business process management common body of knowledge*. Scotts Valley: CreateSpace Independent Publishing Platform.

Suggested Readings

1. Paul Harmon. (2007). *Business process change: a guide for business managers and BPM and six sigma professionals* (2nd ed.). Burlington: Morgan Kaufmann Publishers.
2. Jörg Becker, Martin Kugeler and Michael Rosemann, Process. (2011). *Management: a guide for the design of business processes,* (2nd ed.). New York: Springer.

ITEC-7402 Distributed Database Systems 3(3+0)

A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sited that don't share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database. The increased capabilities of a collection of logically interrelated databases distributed over a computer network enable scalable data processing. This course addresses the components of these systems, covering the main topics The aim of the course is to enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficiencies of the centralized database systems; introduce basic principles and implementation techniques of distributed database systems, expose active research issues in distributed database systems.

Contents

1. Distributed Data Processing, Distributed Database Systems, Data Delivery Alternatives, Promises of DDBSs, Complications Introduced by Distribution, Design Issues, Distributed DBMS Architecture.
2. Distributed Database Design: Top-Down Design Process, Fragmentation, Allocation, Data Directory.
3. Database Integration: Bottom-Up Design Methodology, Schema Matching, Schema Integration, Schema Mapping, Data Cleaning.
4. Data and Access Control: View Management, Data Security, Semantic Integrity Control.
5. Overview of Query Processing: Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing.
6. Query Decomposition and Data Localization: Query Decomposition, Localization of Distributed Data.
7. Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.
8. Multidatabase Query Processing: Issues in Multidatabase Query Processing, Multidatabase Query Processing Architecture, Query Rewriting Using Views, Query Optimization and Execution, Query Translation and Execution.
9. Transaction Management: Definition of a Transaction, Properties of Transactions, Types.

Recommended Texts

1. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (Vol. 2). New York: Prentice Hall.
2. Rahimi, S. K., & Haug, F. S. (2010). *Distributed database management systems*: *a practical approach*. Hoboken: John Wiley & Sons.

Suggested Readings

1. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. New York: Prentice-Hall.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

ITEC-7403 Enterprise Data Center Design and Methodology 3(3+0)

As data centers are inevitably growing more complex and larger, it brings many challenges to the deployment, resource management and service dependability. As data center becomes more and more central in the present age of internet communication, both research and operations communities have begun to explore how to better design and manage them. There are some materials providing guideline for data center design. This course provides a scalable, modular methodology for designing data centers of any size and capability Includes design techniques for accurate planning based on data center capacities It covers all aspects of data center design from site selection to network connectivity. The fundamental design principles take a simple, flexible, and modular approach based on accurate, real-world requirements and capacities. This approach contradicts the conventional method of using square footage to determine basic capacities like power and cooling requirements. The course is aimed to introduce students with the essential knowledge about Data Center and methodologies that optimizes availability, scalability, and performance for that environment and provides centralized management services.

Contents

1. Overview of Data Centers
2. Introduction to Data Centers,
3. Application Architecture Models,
4. Data Center Architecture,
5. Data Center Services.
6. Data Center Requirements
7. Data Center Pre-requisites
8. Budget Constraints, Selecting Geographical Location, Retrofitting.
9. Server Architecture
10. Application Architectures
11. Network Infrastructure.
12. Data Center Security
13. Designing the Data Center Infrastructure
14. Integrating Security into the Infrastructure
15. Server Capacity Planning
16. Best Practices in IT

Recommended texts

1. Mauricio Arregoces and Maurizio Portolani. (2003). *Data center fundamentals*, Indianapolis: Cisco Press.
2. UrsHoelzleand Luiz Andre Barroso. (2009). *The datacenter as a computer: An introduction to the design of warehouse-scale machines*, (1st ed.). San Rafael: Morgan and Claypool Publishers.

Suggested Readings

1. Kailash Jayaswal. (2005). *Administering data centers: servers, storage, and voice over IP*, (1st ed.). Hoboken: Wiley.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

ITEC-7404 **Information Security** **3(3+0)**

As IT continues to grow at a rapid pace and plays a significant role in automating the functions of the enterprise, Information Security has taken on an unparalleled significance around the world. This course is based on Introduction to Information Security: History of IS, what is Security, CNSS Security model, Component of Information Systems, Balancing Information Security and Access, Approaches to Information Security Implementation, The Security System Development Life Cycle, Security Professionals and the Organizations, Communities of Interest. Finally, students will discover how organizations manage and prepare for security incidents, disruptions and disasters and how they manage the day-to-day operations of an information security program. When you are finished with this course, you will have the knowledge and understanding of the bigger picture of information security. Students understand of various types of security incidents and attacks, and learn methods to prevent, detect and react incidents and attacks. Students will also learn basics of application of cryptography which are one of the key technologies to implement security functions.

Contents

1. The Need for Security: Business Need, Threats, Attacks, Secure Software Development. Legal, Ethical, and Professional Issues in Information Security: Laws and Ethics in Information Security, International Laws and Legal Bodies, Ethics and Information Security, Codes of Ethics and Professional Organizations, International Agencies.
2. Risk Management: Introduction to RM, Risk Identification, Risk assessment, Risk Control Strategies, Selecting a risk Control Strategy, Quantitative VS qualitative Risk Control Practices
3. Planning for Security: Information Security Planning and Governance, Information Security Policies, Standards, and Governance, Information Security Blue Prints, Continuity Strategies. Security Technology: Firewalls, VPNs, and Wireless: Access Control, Firewalls, Protecting Remote Connections. Security Technology: Intrusion Detection and Prevention Systems and Other Security Tools: Intrusion Detection and Presentation Systems, Honeyspots, Honeynets
4. Cryptography: Foundations of Cryptography, Cipher Methods, Cryptographic Algorithms, Cryptography Tools, Protocols for secure Communications, Attack on Cryptosystems.Physical Security: Physical Access Controls, Fire Security and Safety
5. Implementing Information Security: Information Security Project Management, Technical Aspects of Implementation, Nontechnical Aspects of Implementation, Information Security Certifications.

Recommended Texts

1. Michael E. Whitman and Herbert J. Mattord. (2011). *Principles of information security*, (4th ed.). San Francisco: Course Technology.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Readings

1. James M. Stewart, Mike Chapple, and Darril Gibson. (2012). *Information systems security professional study guide*, (6th ed.). Hoboken: Sybex.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

1.

ITEC-7405 Knowledge Management Systems & Technologies 3(3+0)

Knowledge Management (KM), is an integrated interdisciplinary presentation that makes sense of the confusingly wide variety of computer science and business KM perspectives arising simultaneously from artificial intelligence, information systems, and organizational behavior. This course examines the characteristics and applications of systems that support knowledge management at personal, group and corporate levels. It considers the electronic representation of knowledge, the components of systems that embody or support knowledge processing and the use and value of such systems. The issue of knowledge management systems has probably always been the most discussed and debated topic within knowledge management. This course will discuss the theoretical implementation of knowledge management systems and its impact on the organization. By the end of the course, students should be able to: Appraise current thought on knowledge management in the light of contemporary debates on knowledge productivity, strategic capability and organizational learning ;Apply theories of knowledge management relevant to current workplace practice; Apply the tools and techniques of knowledge management.

Contents

1. History and paradigms of knowledge management;
2. Types of knowledge,
3. Knowledge Economy,
4. Knowledge Management.
5. KM Processes
6. KM Frameworks and Models
7. KM Frameworks and Models
8. Knowledge Capture and Codification
9. Knowledge Sharing and Communities of Practice
10. Knowledge Application
11. The Role of Organizational Culture
12. Knowledge Management Tools
13. Knowledge Management Strategy
14. The Value of Knowledge Management
15. Organizational Learning and Organizational Memory
16. The KM Team

Recommended Texts

1. KimizDalkir. (2011). *Knowledge management in theory and practice*, (3rd ed.). Cambridge: The MIT Press.
2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

Suggested Reading

1. Elie Geisler and Nilmini Wickramasinghe (January 15, 2009) Principles of knowledge management: theory, practice and cases. New York: M.E. Sharpe.
2. Murray E. Jennex (August 10, 2007) Knowledge management: concepts, methodologies, tools and applications (6-volume set). Hershey: IGI Global.

ITEC-7406 **Multimedia Databases** **3(3+0)**

The spatial, temporal, storage, retrieval, integration and presentation requirements of multimedia data differ significantly from those for traditional data. A multimedia database management system provides for the efficient storage and manipulation of multimedia data in all its varied forms. We look into the basic nature of multimedia data, highlight the need for multimedia DBMSs, and discuss the requirements and issues necessary for developing such systems. Querying a multimedia database consists of retrieving text, images, video, and audio simultaneously. Because the data in a multimedia database are usually not as structured as the data in a conventional database, querying multimedia data is not a trivial problem and has been a popular research area. This course aims to provide a basic study of the development of fundamental multimedia database systems, as well as applicable technologies for developing web-based multimedia applications. The former provides a basis for understanding the basic concepts and techniques pertinent to multimedia databases. In this course, we examine the aspects regarding building multimedia database systems and give an insight into the used techniques. The course deals with content-specific retrieval of multimedia data.

Contents

1. Introduction to Multimedia Databases
2. Multimedia Data
3. An Introduction to SQL and Multimedia
4. Querying Multimedia Data
5. Multimedia Database Architecture and Performance
6. Dealing with Text Databases
7. Dealing with Video Databases.

Recommended Texts

1. Dunckley, L. (2003). Multimedia databases: *An object relational approach*. Boston: Addison-Wesley Longman Publishing Co., Inc.
2. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

Suggested Readings

1. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

2. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

ITEC-7407 Service-Oriented Architecture 3(3+0)

Service-oriented architecture (SOA) emerged in the early part of this century as an evolution of distributed computing. Before SOA, *services* were understood as the end result of the application development process. Before SOA emerged in the late 1990s, connecting an application to data or functionality housed in another system required complex point-to-point integration—integration that developers had to recreate, in part or whole, for each new development project. Exposing those functions through SOA eliminates the need to recreate the deep integration every time. In SOA, the application itself is composed of services. Services can be delivered individually or combined as components in a larger, composite service. By the end of this course students will be able to understand SOA, Service Orientation and Web Services. They will also be able to analyze and design business based on SOA principles. After successful completion of course student will be able to describe SOA (Service-Oriented Architecture) to structure web-based systems, explain WS* services (i.e., SOAP over HTTP, WSDL, UDDI, BPEL), apply REST architecture (i.e., JSON over HTTP, URI) and identify REST design principles.

Contents

1. Introducing SOA
2. Web Services & Primitive SOA
3. Web Services & Contemporary SOA (Activity Management & Composition)
4. Principles of Service-Orientation
5. Service Orientation & Contemporary SOA
6. SOA Delivery Strategies SOA Delivery Lifecycle Phases, Top Down Strategy, Bottom Up Strategy, Agile Strategy.Service Oriented Analysis: Introduction, Benefits of a Business Centric SOA, Deriving Business Services, Deriving Business Services.
7. Service Modelling (Process): Introduction, Modelling Guidelines, Classifying Service Model Logic, Different Modelling Approaches.
8. Service Oriented Design: Introduction, WSDL-Related XML Schema Language, WSDL Language(Like Definitions, Types, Message & Part, Porttype, Interface, & Operation Elements, Input & Output-When Used With Operation, Binding, Input & Output-When Used With Binding, Service, Port & Endpoint, Import, Documentation), SOAP Language Basic (Envelop, Header, Body, Fault) Service Interface Design Tools.
9. Service Oriented Design Steps To Composing SOA, Considerations For Choosing Service Layers, Considerations For Positioning Core SOA Standards (Industry Standards, & SOA, XML & SOA, The WS-I Baisc Profile, WDSL & SOA, XML Schema & SOA, SOAP & SOA, Namespaces & SOA, UDDI & SOA) Considerations For Choosing SOA Extensions (Choosing SOA Characteristics, Choosing WS* Specifications, WS-BPEL & SOA).

Recommended Texts

1. Erl, T. (2005) *Service-oriented architecture (SOA): concepts, technology, and design,* New York: Pearson Education Incorporated.

Suggested Readings

3. Nwosu, K. C., Thuraisingham, B., & Berra, P. B. (2012). *Multimedia database systems: design and implementation strategies*. New York: Springer Science & Business Media.

4. Andries Engelbrecht, (2007) *Computational intelligence: an introduction.* Hoboken: Wiley.

ITEC-7408 **Telecom Management** **3(3+0)**

Telecom Management generally focuses on the basics of Telecom Management. This course helps students to acquire essential skills to manage the vast network of telecommunication. Telecom Management involves both traditional Information Technology and telecommunication lines, delineation between the two has become blurred as the two technologies have increasingly intermingled to effectively send the message. A programme in telecom management includes conceptualizing innovative ideas to manage operations and marketing of telecommunications-related technologies. The purpose of this course is to provide an understanding of the key technical, managerial and policy issues in the effective development and use of telecommunication solutions by organizations. Discussion of technology and technology trends will be set in the context of applications. The interdisciplinary programme offers necessary skills to manage telecom networks – voice, video and data. The course provides the understanding of the operation and management of a telecommunication business.

Contents

1. Introduction to Telecommunication
2. Management and Planning
3. Telecommunication Strategic Planning
4. Developing Requirements and Specifications
5. Managing Long-Distance Services
6. Managing Internet Services
7. Managing PBX and Key Telephone Equipment
8. Managing Automatic Call Distributors
9. Managing Voice Processing Equipment
10. Managing Local Area Network and Internets
11. Managing Wide Area Networks
12. Managing Video and Audio-Conferencing Equipment
13. Managing Convergence
14. Developing a Telecommunication Operations Plan
15. Disaster Prevention and Recovery

Recommended Texts

1. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.
2. P. J. Louis. (2002). *Telecom management crash course: a telecom company survival guide*, (1st ed.). New York: McGraw-Hill Professional.

Suggested Readings

1. Ugarkar. J. (2010*). The essentials of telecommunications management: A simple guide to understanding a complex industry*, Bloomington: AuthorHouse.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hil

ITEC-7409 **Telecommunication Systems** **3(3+0)**

Telecommunications systems include wired and wireless local and wide area networks and hardware and software providing the capabilities for systems to communicate with each other or with users. The set of telecommunications systems supporting most federal government agencies includes network infrastructure and other technical solution components owned by commercial telecommunications service providers and managed on behalf of the government. Telecommunications, also known as telecom, is the exchange of information over significant distances by electronic means and refers to all types of voice, data and video transmission. This is a broad term that includes a wide range of information transmitting technologies such as telephones (wired and wireless), microwave communications, fiber optics, satellites, radio and television broadcasting, the internet and telegraphs. The modern telecommunications infrastructure—made possible by research performed over the last several decades. The course provides an in-depth investigation of telecommunications system concepts and Terminology. The course covers the most popular telecommunication networks and forthcoming technologies.

Contents

1. Communication Channel and the Communication Network
2. Digital Telephony: From Analog to Digital, Advantages and Disadvantages of Digital Techniques, PCM Techniques and Standards, Low Bit-Rate Voice Coding, Voice over the Internet, Devices for Digital Telephony.
3. Switching and Signaling Systems Switching, Circuit Switching and Packet Switching, Signaling Systems, Digital Exchanges, Private Branch Exchanges and Key Phone Systems, Merging Data and Voice, Packet Switching Systems, Switching Fabric Interfaces and ICs, Optics and the Future.
4. Cellular Systems
5. Fixed Wireless Access
6. Digital Subscriber Loop
7. Transmission Techniques.
8. Telecommunication Systems Testing
9. Embedded Systems Design for Telecommunications

Recommended Texts

1. Niha Kularatna and Dileeka Dias. (2004). *Essentials of modern telecommunications systems*, London: Artech House.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

Suggested Readings

1. Lillian Goleniewski. (2002). *Telecommunications Essentials: The Complete Global Source for Communications Fundamentals, Data Networking and the Internet, and Next-Generation Networks,* (1st ed.). Boston: Addison-Wesley Professional.
2. Michael Fitz. (2007). *Fundamentals of Communications Systems*, (1st ed.). New York: McGraw-Hill Professional.

ITEC-7410 Virtualization in Cloud Computing 3(3+0)

The term 'Virtualization' can be used in many respects of computer. It is the process of creating a virtual environment of something which may include hardware platforms, storage devices, OS, network resources, etc. The cloud's virtualization mainly deals with the server virtualization Virtualization in Cloud Computing is making a virtual platform of server operating system and storage devices. This will help the user by providing multiple machines at the same time it also allows sharing a single physical instance of resource or an application to multiple users. This course gives an inside look as traditional physical data centers evolve and morph into virtual entities and cloud environments. Students will learn in-depth details and considerations for planning, designing, and migrating to Virtualized Data Centers (VDCs) and cloud environments. They will learn to design VDC and cloud infrastructures, maintaining the most robust and elastic computing, network, and storage environments, cloud capacity management.

Contents

1. Virtualization: Virtualization Types, Virtualization Management, Cloud Computing, Service Models, Cloud Adoption and Barriers, Return on Investment and Cloud Benefits; Typical Design Patterns and Use Cases, Design Patterns, Cloud Use Cases, Deployment Models, IaaS as a Foundation, Cloud Consumer Operating Model.
2. Cloud Management Reference Architecture: Standards, TMF eTOM, IT Infrastructure Library (ITLI), ITU-T TMN, Building Cloud Models Using Standards, Integration of Management Systems/Functions.
3. Service Fulfillment, Service Assurance, Technical Building Blocks of IaaS: IaaS Service Composition, Developing and Offering Cloud Products, Persisting Service Data.
4. Automating and Orchestration Resources: Building the Cloud, Adding Services to the Cloud, Creation and Placement Strategies, Service Life Cycle Management.
5. Cloud Capacity Management: Tetris and the Cloud, Cloud Capacity Model, Demand Forecasting, Procurement in the Cloud, Case Study - Hybrid Cloud

Recommended Texts

1. VenkataJosyula, Malcolm Orr and Greg Page. (2011). *Cloud Computing: Automating the Virtualized Data Center*, (1st ed.). Indianapolis: Cisco Press.
2. Mickey Iqbal, MithkalSmadi, Chris Molloy, and Jim Rymarczyk. (2011). *IT virtualization best practices: A lean, green virtualized data center approach*, (1st ed.). Chicago: MC Press.

Suggested Readings

1. Aidan Finn, Hans Vredevoort, Patrick Lownds, and Damian Flynn. (2012). *Microsoft private cloud computing*, (1st ed.). Hoboken: Sybex.
2. David Marshall, Wade A. Reynolds, and Dave McCrory. (2006). *Advanced server virtualization: VMware and microsoft platforms in the virtual data cente*r, (1st ed.). Boca Raton: Auerbach Publications.

CSEC-7401 Advanced Artificial Intelligence 3(3+0)

This course will introduce the key principles in artificial intelligence. It will cover simple representation schemes, problem solving paradigms, constraint propagation, and search strategies. Areas of application such as knowledge representation, natural language processing, expert systems, vision and robotics will be explored. The Prolog programming language will also be introduced. Upon successful completion of this course, student will be able to learn and apply searching techniques, learning techniques, identify the major approaches to AI, explain the differences between various types of logic and basic statistical tools used in AI, list the most common methods of statistical learning and classification and explain the basic differences between them, describe the components of Turing machine. At the end, student would be able to: developing their ability to survey existing research and read research papers critically; developing their ability to devise and run experiments, and analyze their results; refining their communication skills; in particular, the ability to present research work to an audience, and to discuss technical material with peers

Contents

1. Introduction: Turing Test, Strong AI vs Weak AI, Heuristics, Applications and Methods, History of AI.
2. Uninformed Search: Search in IS, Genrate-and-Test Paradigm, Blind Search Algorithm.
3. Informed Search: Heuristics, the Best-First Search, The Beam Search, The A* Search, The Bidirectional Search.
4. Search Using Game: Game Trees and Minimum, Game Theory.
5. Logic in AI: Logic and Representation, Propositional Logic, Predicate Logic, Other Logics.
6. Knowledge Representation: Search Tree, Production System, Objects, Frames, Scripts & the Conceptual Dependency System, Semantic Networks, Recent Approaches, Agents.
7. Prolog Programming.
8. Production Systems: Strong Methods vs. Weak Methods, Production System and Inference Methods, Stochastic Processes and Markov Chain.
9. Uncertainty in AI: Fuzzy Sets, Fuzzy Logic, Fuzzy Inference, Probability Theory and Uncertainty.
10. Expert Systems: Characteristics of ES, Knowledge Engineering, Knowledge Acquisition, Classical ES, Case-Based Reasoning.
11. Neural Networks: Introduction, The Perceptron Learning Rule, Back propagation
12. Evolutionary Computation: Simulated Annealing, Genetic Algorithms.

Recommended Texts

1. Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q., & Lee, I. *(2018). Artificial intelligence in the 21st century.* Piscataway: *IEEE* Access.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

Suggested Readings

1. M. Tim Jones. (2008). *Artificial Intelligence: a systems approach*, (1st ed). Burlington: Jones and Bartlett Publishers, Inc.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

CSEC-7402 **Advanced Human Computer Interaction** **3(3+0)**

The aim is to identify the important research issues, and to ascertain potentially fruitful future research directions in relation to the multimodal emotion analysis and to human-computer interaction. Upon successful completion students will be able to explain and reflect critically with the use of course literature on the idea of organizational usability and UX, outline a plan for how to use HCI to create value for and empower the employee/customer/citizen, apply findings from empirical work on HCI issues in the use of multiple organization-wide systems, describe and explain an issue with legacy systems (old systems) using literature from the course and develop and present a comprehensively set of documented and motivated prototypes, sketches, templates, running systems, or scripts that supports HCI interactions within or across organizations, and explain the principle idea behind. Recent research topics will also be discussed in the course. At the end of the course students will be able to identify novel and significant open research questions in electrical engineering and computer science and are able to situate such questions in the contexts of current research literature.

Contents

1. Overview of advanced topics of human-computer interaction,
2. Design thinking and the basic practices of interaction design.
3. Importance of human-computer interaction/interface design
4. Iterative design
5. Input/output techniques
6. How to design and evaluate interfaces
7. Reasearch topics.

Recommended Texts

1. Henderson, A. (2002). *Interaction design: beyond human-computer interaction*. London: Ubiquity.
2. Robertson, J., & Kaptein, M. (Eds.). (2016). *Modern statistical methods for HCI*. New York: Springer.

Suggested Readings

1. Rodrigues, J., Cardoso, P., Monteiro, J., & Figueiredo, M. (2016). *Handbook of research on human-computer interfaces, developments, and applications*. Hershey: IGI Global.
2. Picard, R. W. (2000). *Affective computing*. Cambridge: The MIT press.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: strategies for effective human-computer interaction*. New York: Pearson.
4. Dix, A., Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-computer interaction*. New York: Pearson Education

CSEC-7403 **Advanced Web development** **3(3+0)**

To introduce students with Web Systems and Technologies Implement HTML5 coding solutions using HTML5 elements, attributes and values. Apply CSS33 functionality to Web documents using various properties, selectors and techniques. Integrate basic JavaScript coding into a Web page to create HTML5 APIs. Consider HTML5 and CSS33 techniques to apply to both traditional and mobile delivery platforms, including mobile apps. Transform traditional Web pages into mobile Web pages. Understanding PHP frameworks and advanced technologies. Students will learn the fundamentals of JavaScript code, and then get into jQuery. jQuery is an industry standard framework that lets students quickly and easily write powerful JavaScript. Students will learn how to use some popular jQuery plugins, and gain an understanding of how plugins work, so you can use any plugi Something new in JQuery. Mobile websites and mobile apps development. Upon successful completion of course students will be able to apply learnt PHP, CSS33, boot strapping and WSDL along with advanced JQuery concepts.

Contents

1. Advanced HTML5 concepts: Structure Elements, Validating HTML5 Code, The <video> Element, The <audio> Element, Introduction to HTML5 Forms, Cross-Browser Compatible HTML5 Forms, New Form Elements in HTML5, HTML5 Global Attributes for Form Elements.
2. HTML5 APIs, Document Object Model (DOM), The Canvas API, The Offline AppCache API, The Geolocation API, The Drag-and–Drop API, The File API, The History AP
3. Advanced CSS33 concepts: Cascading Style Sheets (CSS3), Style Guides, CSS3 and HTML, CSS3 Terms and Syntax, Applying CSS3 Styles, Page Layout with CSS3, CSS3 Positioning Schemes
4. Introduction to CSS33, CSS33 Selectors and Properties, CSS33 Background Properties, CSS33 Border Properties, CSS33 Font Properties, CSS33 Text Effects,
5. Introduction to Advanced CSS33 Techniques, CSS33 2D and 3D Transformations, CSS33 Transitions, CSS33 Animations, CSS33 User Interfaces, Creating Menus and Buttons with CSS33.
6. Advanced JavaScript concepts: Learning JavaScript Design Pattern
7. Advanced PHP concepts: functions, date and time, debugging and logging.
8. PHP extensions and application repository –PEAR, Smarty Templates, code reuse, feeding a function, return data from a function, Runtime error handling using PHP exception handling mechanism
9. Web services including SOAP, NU SOAP and WSDL, popular frameworks including CakePHP, Yii, Zend and Codelgniter. Advanced JQuery concepts: Bind/Unbind, Live/Die

Recommended Texts

1. Castro, E., & Hyslop, B. (2015). *HTML5 a CSS3*. Prague: Albatros Media.
2. Negrino, T, & Smith, D. (2013). *Dreamweaver CC: visual quickstart guide*. San Francisco: Berkeley Peachpit Press.

Suggested Readings

1. Pollock, J. (2001). *JavaScript: a beginner's guide*. New York: McGraw-Hill
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

CSEC-7404 **Advanced Web Engineering** **3(3+0)**

Web Engineering introduces a structured methodology utilized in software engineering to Web development projects. The course addresses the concepts, methods, technologies, and techniques of developing Web sites that collect, organize and expose information resources. Advanced Web Engineering can provide you with an unprecedented level of service and expertise for all of your marketing campaigns like email and social media marketing, event management, along with the ability to create offers and local deals for your business. It is a study of the concepts, principles, techniques and methods of Web engineering. The course is aimed to provide students with conceptual understanding required to develop web applications and web services according to international standards**.** It is an extended version of Web Engineering with some material covered in greater depth. By the end of this course students will be able to analyze and design comprehensive systems for the creation, dissemination, storage, retrieval, and use of electronic records and documents.

Contents

1. The Need for Web Engineering
2. Web Effort Estimation
3. Web Productivity Measurement and Benchmarking
4. Web Quality, Web Usability
5. Web System Reliability and Performance, Web Application Testing
6. An Overview of Process Improvement in Small Settings: Initiating SPI Efforts, Process Improvement Cycle, Process Assessments, Implementation in Small Settings.
7. Conceptual Modeling of Web Applications: The OOWS Approach: Introduction, A Method to Model Web Applications, A Strategy to Develop the Web Solution, Case Study: *Valencia CF Web Application.*
8. Model-Based Web Application Development: The OOHDM approach, Building an Online CD Store with OOHDM, From Design to Implementation, Discussion and Lessons Learned.
9. W2000: A Modeling Notation for Complex Web Applications: Introduction, Modeling Elements, Models, Example Application: Information Model, Navigation Model, Presentation Model, 4 Service Model.

Recommended Texts

1. Roger Pressman and David Lowe. (2008). Web engineering: A practioner's approach, (1st ed.). New York: McGraw-Hill, Science/Engineering/Math.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

Suggested Readings

1. Emilia Mendes and Nile Mosley. (2010). Web engineering, (1st ed.). New York: Springer; Softcover reprint of hardcover.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

CSEC-7405 **Agent Based Modeling** **3(3+0)**

Agent Based Models are computer models that attempt to capture the behavior of individuals within an environment. We define traits and initial behavior rules of an agent that organize their actions and interactions. Stochasticity plays an important part in determining which agents interact and how agents make decisions. Students will be able to learn agent-based modelling and social network theory along with biological systems. Because agents carry knowledge with them as they move, they can utilize that knowledge and history as they travel around experiencing new areas and meeting other agents. Students will learn how to build a model from the ground up and how to analyze and understand the results of a model using the NetLogo programming language. We will also discuss how to build models that are sound and rigorous, a self-forming neighborhood model, cellular automata, and current research in this area. By the end of this course, you should be able to think coherently about causation in a complex system, and to communicate those thoughts in writing and speech.

Contents

1. Introduction to agent based modeling.
2. Introduction to Net Logo.
3. Complexity in Social Worlds.
4. Net Logo Commands.
5. Net Logo Procedures.
6. Model properties (Why agent-based objects?
7. Agents, environments, and timescales).
8. Biological systems: fireflies, flocking, slime mold, bees, ants (flocking behavior slime mold).
9. Biological systems: predator/prey, debugging (Verification and validation).
10. Social systems: segregation, Schelling, Micro motives and Macro behavior.
11. A self-forming neighborhood model.
12. Cellular automata.
13. Critical phenomena. Sand piles.
14. Current research topics in Agent Based Modeling.

Recommended Texts

1. Nigel Gilbert. (2008). Agent-based models, Thousand Oaks: SAGE Publications.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

Suggested Readings

1. Emilia Mendes and Nile Mosley. (2010). Web engineering, (1st ed.). New York: Springer; Softcover reprint of hardcover.
2. James Harry Green. (2001). *The irwin handbook of telecommunications management*, (3rd ed.). New York: McGraw-Hill.

CSEC-7406 **Big Data Analytics** **3(3+0)**

The main aim of the course is to frame big data jobs as Apache Spark Scripts. Students should be capable enough to optimize an algorithm by partitioning it across worker nodes. At the end of this course, student will become familiar with the fundamental concepts of Big Data management and analytics. Also, they will become competent in developing distributed code using Scala language and Apache Spark API. The course will provide enough concepts to deploy a spark cluster and run jobs on the cluster. Course will also cover GraphX, MLib, SparkSQL libraries to study their application in social networks and link prediction. This course is for those new to data science. No prior programming experience is needed, although the ability to install applications and utilize a virtual machine is necessary to complete the hands-on assignments. This course focuses on RDD based algorithm design, with an emphasis on text processing algorithms common in natural language processing, information retrieval, and machine learning.

Contents

1 Big Data: Issues and challenges
2 Big Data Tools: An overview
3 Functional Programming Paradigm
4 Scala: Basics, Functions, and Data Structures
5 Scala Practical Exercise
6 Apache Spark & RDD
7 Deploy Apache Spark Cluster
8 Implement and Run Apache Spark Job on Cluster
9 Broadcast Variables and Accumulative Variables in Apache Spark
10 Island, Global, and Grid Models
11 Translating Embeddings for Modeling Multi-relational Data
12 Scalable Genetic Algorithms using Apache Spark
13 ACO for Link Prediction
14 Using Neighborhood for Link Prediction Problems
15 GraphX, MLib, SparkSQL

Recommended Texts

1 Kenneth Cukier and Viktor Mayer-Schönberger (2014). Big data: a revolution that will transform how we live, work, and think. London: Eamon Dolan/Mariner Books
2 Jimmy Lin and Chris Diyer (2010). Data-intensive text processing with mapreduce. California: Morgan & Claypool Publishers

Suggested Readings

1 Andy Konwinski, Holden Karau, Matei Zaharia, and Patrick Wendell (2015). Learning spark: lightning-fast big data analysis. Sebastopol: O'Reilly Media.
2 Anand Rajaraman and Jeff Ullman (2011), *Mining of massive dataset*s. Cambridge: Cambridge Press.

CSEC-7407 **Bioinformatics** **3(3+0)**

This course aims to introduce students to the current bioinformatics concepts and their implementations. A student completing a major in Bioinformatics shall be able to apply knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics, existing software effectively to extract information from large databases and to use this information in computer modeling, problem-solving skills, including the ability to develop new algorithms and analysis methods, an understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory. Scholars are able to learn and understand Distance-Based Methods of Phylogenetics, Phylogenetic Trees, Terminology of Tree Reconstruction, Rooted and Unrooted Trees, Gene Vs. Species Trees, Character and Distance Data. Distance Matrix Methods UPGMA. Estimation of Branch Lengths, Transformed Distance Method, Neighbor's Relation Method, Neighbor-Joining Methods. Maximum Likelihood Approaches. Multiple Sequence Alignments. Term papers are part of course.

Contents

1. Molecular Biology and Biological Chemistry: The Genetic Material: Nucleotides, Orientation, Base Pairing, Central Dogma of Molecular Biology. Gene Structure and Information Content: Promoter Sequences, The Genetic Code. Introns and Exons. Protein Structure and Function: Primary, Secondary, Tertiary and Quaternary Structure. Nature of Chemical Bonds: Anatomy of Atom, Valence. Electronegativity, Hydrophilicity and Hydrophobicity. Molecular Biology Tool.
2. Data Searches and Pairwise Alignments: Dot Plots. Simple Alignments. Scoring. Gaps: Simple Gap Penalties, Origination and Length Penalties. Scoring Matrices. Dynamic Programming: The Needleman AndWunsch Algorithm. Local and Global Alignments: Global and Semi-Global Alignments, The Smith-Waterman Algorithm. Database Searches: BLAST and Its Relatives, Other Algorithms.
3. Substitution Patterns: Patterns of Substitutions Within Genes: Mutation Rates. Functional Constraint. Synonymous Vs. Non-synonymous Changes, Indels and Psuedogenes, Substitutions Vs. Mutations, Fixation. Estimating Substitution Numbers: Jukes/Cantor Model, Transitions and Transversions. Kimura's Two-Parameter Model, Models with Even More Parameters, Substitutions Between Protein Sequences. Variations in Substitution Rates Between Genes. Molecular Clocks.
4. Character-Based Approaches to Phylogenetics: Parsimony: Informative and Uninformative Sites, Unweighted Parsimony, Weighted Parsimony. Inferred Ancestral Sequences. Searching Strategies.

Recommended Texts

1. Krane, D. E. (2002). *Fundamental concepts of bioinformatics*. New York: Pearson Education.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

Suggested Readings

1. Andy Konwinski, Holden Karau, Matei Zaharia, and Patrick Wendell (2015). Learning spark: lightning-fast big data analysis. Sebastopol: O'Reilly Media.

2. Anand Rajaraman and Jeff Ullman (2011), *Mining of massive datasets*. Cambridge: Cambridge Press.

CSEC-7408 Cloud Computing 3(3+0)

Cloud Computing has transformed the IT industry by opening the possibility for infinite or at least highly elastic scalability in the delivery of enterprise applications and software as a service (SaaS). Amazon Elastic Cloud, Microsoft's Azure, Google App Engine, and many other Cloud offerings give mature software vendors and new start-ups the option to deploy their applications to systems of infinite computational power with practically no initial capital investment and with modest operating costs proportional to the actual use. This course gives an introduction to cloud computing and related techniques, issues, ecosystem and case studies. Students will learn and understand about such fundamental distributed computing "concepts" for cloud computing, how these techniques work inside today's most widely-used cloud computing systems and various research papers will be studied. Students will learn Datacenter Architectures, Cloud Stack , Technology Trends, Consistency, Availability, Partitions, Cluster File Systems, Data-flow Computation Frameworks, Key-Value Store and Interactive Query Systems, Big Data in the Clouds, Geographic distributed Storage, Programming Languages for the Cloud, DBases in the Cloud, In-Memory Frameworks, Google file system, Hadoop file system, MapReduce, OSes and Clouds Networking: topologies, Networking: Traffic Management, Networking: Transport Protocol Improvements, Security, Scheduling and Resource Management in clouds, Software Level Agreements.

Contents

1. Overview of Distributed Computing, Emergence of Cloud Computing, Global Nature of the Cloud, Cloud-Based Service Offerings, Grid Computing, Reliability of Cloud Model, Benefits of Cloud Model, Legal Issues,
2. Key Characteristics of Cloud Computing, Challenges for the Cloud. The Evolution of Cloud Computing.
3. Web Services Delivered from the Cloud: Communication-as-a-Service (CaaS), Infrastructure-as-a-Service, Monitoring-as-a-Service (MaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS),
4. Storage: HDFS, NAAS, SAN, Distributed Graph Processing, MapReduce, MapReduce Paradigm, MapReduce Examples, Emerging Paradigms, Distributed Graph Processing, Hadoop Scheduling, Dominant-Resource Fair Scheduling, Building Cloud Networks. Virtualization. Federation, Presence, Identity, and Privacy in the Cloud. Security in the
5. Cloud. Common Standards in Cloud Computing. End-User Access to Cloud Computing. Mobile Internet Devices and the Cloud.

Recommended Texts

1. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. Boca Raton: CRC press.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

Suggested Readings

1. Linthicum, D. S. (2009). *Cloud computing and SOA convergence in your enterprise: a step-by-step guide*. New York: Pearson Education., Boston: Addison-Wesley Professional.

2. Sosinsky, B. (2010). *Cloud computing bible* (vol. 762). Hoboken: John Wiley & Sons.

CSEC-7409 **Computational Intelligence** **3(3+0)**

The course is aimed to provide students' knowledge of up-to-date efficient variants of different meta heuristics together with their potential applications. Also, to develop the interest of students in both research and development. The course goal is to make students familiar with basic principles of various computational methods of data processing that can commonly be called computational intelligence (CI). Here belong mainly bottom-up approaches to solutions of (hard) problems based on various heuristics (the so-called soft computing), rather than exact approaches of traditional artificial intelligence based on logic (hard computing). Examples of CI are nature-inspired methods (neural nets, evolutionary algorithms), fuzzy systems, as well as various probabilistic methods under uncertainty (e.g. Bayesian models) and machine learning methods (e.g. reinforcement learning). After the course the students will be able to conceptually understand the important terms and algorithms of CI, such that they would be able to choose appropriate method(s) for a given task.

Contents

1. Introduction to Computational Intelligence topics, Fundamental concepts
2. Introduction to Evolutionary Computing, Evolutionary computation terms, Canonical genetic algorithm
3. Evolutionary computation variations, Evolutionary programming and Evolution strategies
4. Implementation of Evolutionary Computing
5. Swarm Intelligence, Particle Swarm Optimization
6. Classification, Learning, and Adaptation, Supervised, Unsupervised, Reinforcement Learning
7. Data partitioning and Cross Validation, Error metrics: Mean squared error, receiver operating characteristic curves, Neural Networks and Evolutionary Computation: Explanation and Sensitivity Analysis
8. Neural Networks Implementation, Evolutionary and swarm-based neural networks
9. Fuzzy sets, Membership functions, linguistic variables
10. Fuzzy Logic, Fuzzy set operators, Fuzzy rule-based systems
11. Fuzzification, defuzzification, Fuzzy control, Evolving fuzzy rule systems
12. Basic ANN Code, Neuro-fuzzy systems
13. Fuzzy-GA systems, Support Vector Machines
14. Probabilistic reasoning: Bayesian reasoning and Dempster-Shafer theory
15. Bayesian belief networks, Fuzzy belief networks, Evolving belief networks
16. Artificial Immune Systems

Recommended Texts

1. Russell Eberhart and Yuhui Shi. (2007). *Computational intelligence: concepts to implementations*. Amsterdam: Elsevier.
2. Fakhreddine Karray and Clarence de Silva. (2004). *Soft computing and intelligent systems design.* New York: Pearson Edition.

Suggested Readings

1. Amit Konar. (2005). *Computational intelligence: principles, techniques, and applications.* New York: Springer Science & Business Media.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

CSEC-7410 **Computer Vision** **3(3+0)**

Computer Vision is a field of Artificial Intelligence and Computer Science that aims at giving computers a visual understanding of the world. Computer vision spans all tasks performed by biological vision systems, including "seeing" or sensing a visual stimulus, understanding what is being seen, and extracting complex information into a form that can be used in other processes. The goal of Computer Vision is to emulate human vision using digital images through main processing components. The course concentrates on the knowledge of Computer Vision and its importance in computing area having multiple applications. By the end of the course students will be able to develop basic methods for applications that include finding known models in images, depth recovery from stereo, camera calibration, three-dimensional image reconstruction, rendering based on images, Laws' Texture Energy Measure image stabilization, Syntactic Texture Description Methods, automated alignment, tracking, boundary detection, and recognition. On successful completion of this course, students should have the skills and knowledge to understand knowledge, theories and methods in computer vision.

Contents

1. Computer Vision an Introduction.
1. Image formation, Image Processing, Feature Detection and Matching.
2. Feature-based Alignment.
3. Image Stitching.
4. Dense Motion Estimation.
5. Structure from Motion and Recognition.
6. Computational Photography.
7. Stereo Correspondence.
8. 3D Reconstruction.
9. Image-based Rendering.
10. Statistical Texture Description, Methods Based on Spatial Frequencies, Co-occurrence Matrices, Edge Frequency, Primitive Length (Run Length).
11. Laws' Texture Energy Measure
12. Fractal Texture Description, Multiscale Texture Description – Wavelet Domain Approaches, other Statistical Methods of Texture Description
13. Syntactic Texture Description Methods, Shape Chain Grammars, Graph Grammars, Primitive Grouping in Hierarchical Textures, Hybrid Texture Description methods, Texture Recognition Method Applications.

Recommended Texts

1. Richard Szeliski, (2011). *Computer vision algorithms and applications*, New York: Springer.
2. Milan Sonka. (2008). *Image processing, analysis, and machine vision*, (3rd ed.). Washington: CL Engineering

Suggested Readings

1. Dr Simon J. D. Prince. (2012). *Computer vision: models, learning, and inference*. Cambridge: Cambridge University Press.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

CSEC-7411 **Digital Image Processing** **3(3+0)**

This course provides an introduction to basic concepts, methodologies and algorithms of digital image processing including image sampling and quantization, color, point operations, segmentation, morphological image processing, linear image filtering and correlation, image transforms, eigen images, multiresolution image processing, noise reduction and restoration, feature extraction and recognition tasks, image registration. Emphasis is on the general principles of image processing. Students learn to apply material by implementing and investigating image processing algorithms in Matlab. Scholars are able to learn different image processing techniques and compare them. Different recent state of the art articles are given for term paper and the end of semester a term paper is submitted on the following topics Spatial Frequency, Fourier Theory, the Discrete Fourier Transform, Investigating Spectra Filtering of Images, Deconvolution , Storage Media, File Formats, The PBM, PGM, PPM Formats, The Portable Network Graphics (PNG) Format in a group format. Presentation of term paper is also part of course evaluation.

Contents

1. What Is Digital Image Processing?
2. The Origins of Digital Image Processing,
3. Examples of Fields that Use Digital Image Processing.
4. Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition.
5. Spatial Frequency, Fourier Theory, the Discrete Fourier Transform, Investigating Spectra.
6. Filtering of Images, Deconvolution.
7. Color Fundamentals, Color Models.
8. Storage Media, File Formats, The PBM, PGM, PPM Formats, The Portable Network Graphics (PNG) Format.
9. Segmentation Fundamentals, Point Line and Edge Detection.
10. Morphological Image Processing Introduction, Basic Concepts, Fundamental Operations.
11. Compound Operations.
12. Morphological Filtering, Greyscale Morphology.
13. Image Compression Introduction, Redundancy, Performance Characterisation.
14. Lossless Compression Techniques
15. Lossy Compression Techniques, Compression of Moving Images.

Recommended Texts

1. Efford, N. (2000). *Digital image processing: a practical introduction using java* (with CD-ROM). Boston: Addison-Wesley Longman Publishing Co., Inc.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

Suggested Readings

1. Ross, L., & Russ, J. C. (2011). *The image processing handbook.* Cambridge: Microscopy and microanalysis.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

CSEC-7412 **Distributed Computing** **3(3+0)**

This course covers the architecture and enabling technologies of Peer-2-Peer and distributed computing systems and their innovative applications. Students will learn web distributed environment, web services, grid computing, P2P applications protocols, and architectures along with middleware protocols. The course also focuses on Web Services Protocols like SOAP, WSDL, UDDL WS-Extensions, web service deployment along with P2P deployment using Jxta. By the end of course students will be able to understand the principles and desired properties of distributed systems based on different application areas, understand and apply the basic theoretical concepts and algorithms of distributed systems in problem solving, recognize the inherent difficulties that arise due to distributed-ness of computing resources, identify the challenges in developing distributed applications. Students will identify the core concepts of distributed systems: the way in which several machines orchestrate to correctly solve problems in an efficient, reliable and scalable way. Students will examine how existing systems have applied the concepts of distributed systems in designing large systems, and will additionally apply these concepts to develop sample systems.

Contents

1. Introduction to Distributed Systems: Centralized and Decentralized Systems, Taxonomy Dependency Considerations, Examples of Distributed Applications, and Examples of Middleware
2. Structured Document Types: HTML, XML, XHTML, Document Modelling and Validation. Distributed Security Techniques: Introduction, Design Issues, Hash Functions, Signing Messages with a Digital Signature, Secure Channels, Secure Mobile Code: Creating a Sandbox.
3. The Web - Distributed Environment - Naming Things in a Uniform Way, URI Templates, Shared Representation of Things, Hypertext Transfer Protocol, HTTP and Security, Representational State Transfer, The Semantic Web. Peer-2-Peer Environments
4. Web Services
5. Grid Computing
6. P2P Applications Protocols and Architectures: Gnutella: History, Scenario: Discovering Peers, Gnutella in Operation, Searching, Gnutella 0.6 Protocol Description, File Downloads, Implementations. Scalability: Freenet: Introduction, Freenet Routing, Freenet Keys, Joining the Network, Current Developments. BitTorrent: What Is BitTorrent? The BitTorrent Protocol, BitTorrent, Inc., Middleware Protocols, Architectures: Jini: Introdcution, Architecture

Recommended Texts

1. Taylor, I. J., & Harrison, A. (2008). *From P2P and grids to services on the web: evolving distributed communitie s*. New York: Springer Science & Business Media.
2. Oram, A. (2001). *Peer-to-Peer: Harnessing the power of disruptive technologies*. Sebastopol: O'Reilly Media, Inc.

Suggested Readings

1. Ross, L., & Russ, J. C. (2011). *The image processing handbook.* Cambridge: Microscopy and microanalysis.
2. Xiong, J. (2006). *Essential bioinformatics*. Cambridge: Cambridge University Press.

CSEC-7413 Distributed Systems 3(3+0)

To help students gain a general understanding of the principles and concepts governing distributed environment in complex applications. This course covers general introductory concepts in the design and implementation of distributed systems, covering all the major branches such as Cloud Computing, Grid Computing, Cluster Computing, Supercomputing, and Many-core Computing. Distributed Systems combine the computational power of multiple computers to solve complex problems. The students will also learn interprocess communication, distributed file systems, and distributed shared memory. The students ill also learn fault tolerance and recovery techniques. By the end of course students will be able to Apply knowledge of distributed systems techniques and methodologies, explain the design and development of distributed systems and distributed systems applications, multidatabase query processing techniques, transaction management, types and its properties, database integrations, query processing problems and techniques, optimization of distributed queries, use the application of fundamental Computer Science methods and algorithms in the development of distributed systems.

Contents

1. Distributed Data Processing, Distributed Database Systems, Data Delivery Alternatives, Promises of DDBSs, Complications Introduced by Distribution, Design Issues, Distributed DBMS Architecture.
2. Distributed Database Design: Top-Down Design Process, Fragmentation, Allocation, Data Directory.
3. Database Integration: Bottom-Up Design Methodology, Schema Matching, Schema Integration, Schema Mapping, Data Cleaning.
4. Data and Access Control: View Management, Data Security, Semantic Integrity Control.
5. Overview of Query Processing: Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing.
6. Query Decomposition and Data Localization: Query Decomposition, Localization of Distributed Data.
7. Optimization of Distributed Queries: Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.
8. Multidatabase Query Processing: Issues in Multidatabase Query Processing, Multidatabase Query Processing Architecture, Query Rewriting Using Views, Query Optimization and Execution, Query Translation and Execution.
9. Transaction Management: Definition of a Transaction, Properties of Transactions, Types.

Recommended Texts

1. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.
2. Rahimi, S. K., & Haug, F. S. (2010). *Distributed database management systems*: *a practical approach*. Hoboken: John Wiley & Sons.

Suggested Readings

1. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. New

York: Prentice Hall.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7414 Expert Systems 3(3+0)

This is an especially powerful approach for problems that change often or where solutions involve application of human knowledge, rather than intricate calculations. Today, they are used in business, science, engineering, manufacturing, etc. Example applications include: computer configuration, fault diagnosis, computer-aided instruction, data interpretation, planning and prediction, and process control. The course concentrates on an analysis of the architecture, knowledge and problem-solving style of expert system. Students will learn distinguishing features of Expert Systems, theoretical features, basic form of inference, basic components of expert systems, knowledge engineering tools. Students will be able to explain and describe the concepts central to the creation of knowledge bases and expert systems, to apply the tools and the processes used for the creation of an expert system. Student will know methods used to evaluate the performance of an expert system, conduct an in-depth examination of an existing expert system with an emphasis on basic methods of creating a knowledge base. Students will be able to examine properties of existing systems in a case-study manner, comparing differing approaches.

Contents

1. Introduction to Expert Systems.
2. Major Characteristics of Expert Systems.
3. Expert System Architecture, Knowledge Representation.
4. Prolog Programming, Developing Meta Interpreters in Prolog.
5. Inference Techniques, MYCIN.
6. Rule-Based Expert Systems.
7. Backward-Chaining Rule-Based Systems.
8. Designing Backward-Chaining Rule-Based Systems.
9. Forward-Chaining Rule-Based Systems.
10. Designing Forward-Chaining Rule-Based Systems.
11. Frame-Based Expert Systems.
12. Designing a Frame-Based Expert Systems.

Recommended Texts

1. Durkin, J. (1998). *Introduction to Expert Systems*, (3rd ed.). Boston: Addison-Wesley.
2. Ken Pedersen. (1989). *Expert Systems Programming: Practical Techniques for Rule-Based Systems*, (1st ed.). Hoboken: Wiley.

Suggested Readings

1. Patterson, D. W. (1990). *Introduction to artificial intelligence and expert systems*. New York: Prentice Hall.
2. Nikolopoulos, C. (1997). *Expert systems: introduction to first and second generation and hybrid*

knowledge-based systems. New York: Marcel Dekker, Inc.
3. Rich, E., & Knight, K. (1992). *Artificial Intelligence: Instructor's Manual*. New York: McGraw-Hill.
4. Jean-Louis Lauriere. (1990). *Problem Solving and Artificial Intelligence*. New York: Prentice Hall.

CSEC-7415 Formal Methods 3(3+0)

Modern software development inevitably requires the design and analysis of a number of different artifacts. Formal methods allow the mathematically precise formulation of some of these artifacts. This course is an introduction to the use of formal methods for the specification, de-sign, and automatic analysis of software systems. Focus is on theory for specification, validation and verification of network and network-based service functionality specified by communicating state machines, protocol algebraic formulas, and temporal logic descriptions. The theory covers verification by process algebra, temporal logic, rewriting logic and reasoning on UML constraints. With respect to verification tools, model checkers, theorem provers, SAT and SMT solvers will be discussed. More specifically, it deals with formal vs informal requirements, state transition diagrams, predicates, various elements of Z notational language, schema and schema calculus, formal reasoning with famous 8-queen problem, formal intuitions, and other state-of-the-art research in this area.

Contents

1. Formal methods: What is FM? What FM are not, When, How, and Why use FM?
2. Popular Fallacies and Alternatives.
3. Formal Methods and Project Management: Gathering Requirement, From Information Requirement to Formal Specifications, Validating Formal Specifications.
4. Introducing Z: What is Z? Informal Requirements, Data Flow Diagrams, State Transition Diagram.
5. State Transition Diagram, State Transition tables. Introducing schemas: Basic Types and Abbreviations, Axiomatic Descriptions, State Schemas, Operation Schemas, Implicit preconditions
6. Elements of Z: Sets and Types, Declarations, Variables, Expressions, Operators, Predicates, Equations and Laws. Structure: Tuple, Records, Relations, Tables, Databases, Pairs and Binary Relations, Functions, Sequences, Operators.
7. Logic: Basic Predicates, Using Predicates in Z, Relations as Predicates, Logical Connectives, Logic and Natural Language, Quantifiers, Z and Boolean Types, Predicates and undefined Expressions. Synthesis: Set Comprehensives, Lambda Expressions, Formal Specifications
8. Schemas and schema calculus: Conjunctions and Disjunctions, Other Schema Operators. Schema types and Bindings: Generics & Free Types.
9. Formal Reasoning: Calculation and proof, Laws, Checking Specifications, Preconditions
10. Graphical User Interface, Converting Z Specification into Code.

Recommended Texts

1. Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., & Schnoebelen, P. (2013). *Systems and software verification: model-checking techniques and tools*. New York: Springer Science & Business Media.
2. Jacky, J. (1997). *The way of Z: practical programming with formal methods*. Cambridge: Cambridge University Press.

Suggested Readings

1. Diller, A. (1994). Z: an introduction to formal methods. Hoboken: John Wiley & Sons, Inc.

2. Clarke Jr, E. M., Grumberg, O., Kroening, D., Peled, D., & Veith, H. (2018). Model checking. Cambridge: The MIT press.

CSEC-7416 Functional Programming 3(3+0)

Broadly speaking, functional programming is a style of programming in which the primary method of computation is the application of functions to arguments. Among other features, functional languages offer a compact notation for writing programs, powerful abstraction methods for structuring programs, and a simple mathematical basis that supports reasoning about programs. The students will gain an applied understanding of the concepts and practice of functional programming. The goal of the course is to delve deeper in to the principles of program design, implementation and understanding. The course aims to use the language to implement algorithms and data types to solve problems; to explain common errors in sample programs and offer corrections to discuss functional programming and functional programs. On completion of the course, the student should be able to list and define the fundamental concepts of functional programming, manually execute a given (simple) functional program, manually infer the type of a given (simple) functional program, implement (simple) algorithms and data structures as functional programs and design (large) functional programs that are modular etc.

Contents

1. Introducing functional programming: Computers and modeling, Pictures and functions, Types, The Haskell programming language, Haskell Interpreter, Implementing a Prime Number Test,
2. Haskell Type Declarations, Identifiers in Haskell, Haskell Types, The Prime Factorization Algorithm, The map and filter Functions, Haskell Equations and Equational Reasoning.
3. Mathematical Objects: Logical Connectives and their Meanings
4. Lambda Abstraction, Definitions and Implementations, Abstract Formulas and Concrete Structures, Logical Handling of the Quantifiers, Quantifiers as Procedures.
5. The Use of Logical Proof: Proof Style, Proof Recipes, Rules for the Connectives
6. Sets, Types and Lists: Sets, Paradoxes, Types and Type Classes, Special Sets, Algebra of Sets
7. Comprehension and Database Query, Using Lists to Represent Sets, A Data Type for Sets.
8. Relations: The Notion of a Relation, Properties of Relations, Implementing Relations as Sets of Pairs.
9. Implementing Relations as Characteristic Functions, Equivalence Relations
10. Functions: Basic Notions, Surjections, Injections, Bijections, Function Composition, Inverse Function, Partial Functions, Functions as Partitions, Products, Congruences.
11. Induction and Recursion: Mathematical Induction, Recursion over the Natural Numbers
12. Induction and Recursion over Trees, Induction and Recursion over Lists

Recommended Tests

1. Sankel, D. (2006). The haskell road to logic, maths and programming. London: King's College Publications.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. MacLennan, B. J. (1990). Functional programming: practice and theory. Boston: Addison-Wesley Longman Publishing Co., Inc

2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7417 **Fuzzy Logic** **3(3+0)**

This course will start with a brief introduction to fuzzy sets. The differences between fuzzy sets and crisp sets will be identified. Various terms used in the fuzzy sets and the grammar of fuzzy sets will be discussed, in detail, with the help of some numerical examples. It is an important area having multiple applications in Computing. The course will introduce students with key concepts of Fuzzy Logic. Students will learn new paradigm fuzzy access to some part of the classical mathematics and logic. To show how to model fuzzy sets, how to handle with arithmetic of fuzzy quantities, and to acquire operations with fuzzy relations. To understand the mechanism of fuzzy reasoning and the role the essential models in fuzzy inference. Upon successful completion of course students will be able to understand fuzzy set theory, recognize fuzzy logic membership function, recognize fuzzy logic fuzzy inference systems, make applications on Fuzzy logic membership function and fuzzy inference systems, use the fuzzy set theory on the statistical method which is given.

Contents

1. Fuzziness: Introduction, Examples of Fuzziness, Modeling of Fuzziness.
2. Operations on Fuzzy Sets, Fuzziness as Uncertainty
3. Algebra of Fuzzy Sets: Boolean Algebra and Lattices.
4. Equivalence Relations and Partitions, Composing Mappings,
5. Isomorphism, and Homomorphisms, Alpha Cuts, Images of Alpha Level Sets.
6. Fuzzy Quantities, Fuzzy Numbers, Fuzzy Intervals.
7. Basic Connectives: t – Norms, Generators of t – Norms.
8. Isomorphisms of t – Norms, Negations, t – Conorms, Strict De – Morgan Systems, Nilpotent De Morgan Systems, Non-uniqueness of Negations in Strict De Morgan Systems.
9. Fuzzy Implications, Averaging Operators and Negations
10. De Morgan Systems with Averaging Operators, Power of t-Norms, Sensitivity of Connectives.
11. Fuzzy Relations: Binary Fuzzy Relations, Operations on Fuzzy Relations, Approximate Reasoning, Approximate Reasoning in Expert Systems, Modulus Ponens.
12. Universal Approximations: Fuzzy Rule Bases, Desing Methodologies, Approximation Capabilities
13. Partial Knowledge: Belief Functions, Indicence Algebras,
14. Monotonicity, Beliefs, Densities and Allocations, Möbius Transform,
15. Reasoning with Belief Functions, Decision Making with Belief Functions.

Recommended Texts

1. Walker, E., & Nguyen, H. T. (2006). *A first course in fuzzy logic.* Boca Raton: CRC press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Zimmermann, H. J. (2011). *Fuzzy set theory—and its applications*. New York: Springer Science & Business Media.
2. Dubois, D., & Prade, H. (Eds.). (2012). *Fundamentals of fuzzy sets* (vol. 7). New York: Springer Science & Business Media.
3. Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (vol. 4). Hoboken: Prentice hall.

CSEC-7418 **Game Theory** **3(3+0)**

This course provides an introduction to Game Theory. Game Theory is a mathematical framework which makes possible the analysis of the decision-making process of interdependent subjects. It is aimed at explaining and predicting how individuals behave in a specific strategic situation, and therefore help improve decision making. Students will learn how to represent an economic situation as a game and how to analyze it using different equilibrium concepts proposed in the literature, the prominent one being the Nash equilibrium, solving repeated reachability and eventual safety games. The course concentrates on strategic interaction under incomplete information and modify the Nash equilibrium concept to include the uncertainty of the players about some of the parameters of the game. Often, an equilibrium concept fails to provide a unique solution to the game. At the endof this course students will deal with the problem of indeterminacy in games in extensive form and introduce refinements of the Nash equilibrium.

Contents

1. Introduction to Game Theory, The Two-Person, Zero-Sum Game with Equilibrium Points, The General, Two-Person, Zero-Sum Game.
2. Utility Theory, The Two Person, Non-Zero-Sum Game.
3. The N-Person Game
4. Strategic Games: Basic Concepts, Iterated Elimination of Strategies, Mixed Extension.
5. Variations on the Definition of Strategic Games, Mechanism Design.
6. Pre-Bayesian Games
7. Infinite Games and Automata Theory: Basic Notations and Definitions, Transformation of Winning Conditions, Tree Automata, Beyond Finite Automata.
8. Algorithms for Solving Parity Games: Games on Graphs.
9. Solving Repeated Reachability and Eventual Safety Games.
10. Solving Parity Games.
11. Back and Forth Between Logic and Games: Introduction, Reachability Games and Parity Games, Reachability Games and Logic.
12. Logics with Least and Greatest Fixed-Points, Definability of Winning Regions in Parity Games, Inflationary Fixed-Point Logic and Backtracking Games.
13. Graph Searching Games: Introduction, Classifying Graph Searching Games

Recommended Texts

1. Davis, M. D. (2012). Game theory: a nontechnical introduction. Chelmsford: Courier Corporation.
2. Apt, K. R., & Grädel, E. (Eds.). (2011). Lectures in game theory for computer scientists. Cambridge: Cambridge University Press.

Suggested Readings

1. Hearn, R. A., & Demaine, E. D. (2009). *Games, puzzles, and computation*. Boca Raton: CRC Press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7419 **Grammar Engineering** **3(3+0)**

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview, LFG Basics.
3. LFG Basics II, Templates I, MacOSX, Unix.
4. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file
5. Lexical Rules, Passive and Argument alternations.
6. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation
9. Functional Uncertainty, Imperatives and empty categories.
10. Finite-State Morphology (FSM) I.
11. FSM II (-unknown), Free Word Order and the Shuffle Operator.
12. Meta-categories, Meta-rule-macros and Coordination.

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. Stanford: CSLI. Publications.
2. Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., & Newman, P. (2008). *XLE documentation.* California: Palo Alto Research Center.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar*. Leiden: Brill.
2. Dalrymple, M., Kaplan, R. M., Maxwell III, J. T., Maxwell, J. C., & Zaenen, A. E. (1995). *Formal issues in lexical-functional grammar* (No. 47). Stanford: Center for the Study of Language (CSLI).

CSEC-7420 **Graph Theory & Algorithms** **3(3+0)**

This course is aimed to cover a variety of different problems in Graph Theory. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Various graphs algorithms will also be taught along with its analysis. By taking this course, one would be able to master fundamental concepts in Graph Theory, get to know a wide range of different Graphs, and their properties, be able to perform Elementary, Advanced Operations on Graphs to produce a new Graph, understand Graph Coloring, understand Eulerian and Hamiltonian paths and circuits. And many related topics to Paths, know how to turn a Graph into a Matrix and vice versa, obtain a solid foundation in Trees, Tree Traversals, and Expression Trees, have a good understanding of Graph Match. Topics covered in this course include: Fundamental Concepts of Graphs, Sub-Graphs and Super-Graphs, Connected Graphs (Walks and connection), Trees (including forests, spanning trees, Calay's Formula for positive integers, and other applications of trees), Non-Separable graphs and its applications, Connectivity among graphs, Planar graphs, Vertex coloring etc.

Contents

1. Fundamental Concepts of Graphs: What is A Graph, Simple Graphs, Graph and Their Representations, Isomorphism and Automorrphisms,
2. Labelled Graphs, Graphs Arising from Other Structures, Incidents Graphs, Union and Intersection Graphs, Cartesian Product, Directed Graphs.
3. Sub-Graphs: Sub-Graphs and Super graphs, Spanning and Induced Sub-Graphs, Decomposition and Coverings, Edge Cuts and Bonds, Even Sub-Graphs, Graph Reconstruction.
4. Connected Graphs: Walks and Connection, Cut Edges, Connection to Diagraphs
5. Trees: Forests and Trees, Spanning Tree, Calay's Formula, Fundamental Cycles AndBonds, Co-Tree, Trees and Distance. Applications of Tree.
6. Non-separable Graphs: Cut Vertices, Separations and Blocks, Ear Decompositions, Strong Orientations, Directed Ear Decompositions, Even Cycles Decompositions.
7. Connectivity: Vertex Connectivity, Fan Lemma, Edge Connectivity, Three-Connected Graphs
8. Planar Graphs: Plane and Planar Graphs, Duality, Euler's Formula, Bridges, Kuratowski'sTheorem.
9. Vertex Colorings: Chromatic Numbers, Critical Graphs, Girthand Chromatic Number, Perfect Graphs, List Colourings, The Adjacency Polynomial, Chromatic Polynomial.
10. Edge Colourings: Edge Colouring Number, Vizing's Theorem, Snarks
11. Hamilton Cycles: Hamiltonian and non-Hamiltonian graphs
12. Eigenvalues of Graph, Coverings and Packings in Directed Graphs, Integer Flows and Coverings.

Recommended Texts

1. Chartrand, G., & Zhang, P. (2013). A first course in graph theory. Chelmsford: Courier Corporation.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Chartrand, G., Lesniak, L., & Zhang, P. (2010). Graphs & digraphs (Vol. 39). Boca Raton: CRC press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7421 **Information Retrieval** **3(3+0)**

Information retrieval covers the tasks of indexing, searching, and recalling data, particularly text or other unstructured forms. It has an important role to play in a large number of applications viz., digital libraries, office automation, internet and e-commerce. The aim of the course is to study theoretical aspects as well as implementation issues of classical and modern retrieval problems. Web search is the application of information retrieval techniques to the largest corpus of text anywhere — the web — and it is the context where many people interact with IR systems most frequently. In this course, we will cover basic and advanced techniques for building text-based information systems, including the following topics: Efficient text indexing, Boolean and vector-space retrieval models, Evaluation and interface issues, IR techniques for the web, including crawling, link-based algorithms, and metadata usage, Document clustering and classification, Traditional and machine learning-based ranking approaches. Course will also include few practical exercises.

Contents

1. Introduction: Basic Concepts of IR, Data Retrieval & Information Retrieval, IR System Block, Diagram. Automatic Text Analysis, Luhn's Ideas, Conflation Algorithm, Indexing and Index Term Weighing, Probabilistic Indexing, Automatic, Classification. Measures of Association, Different Matching Coefficient, Classification Methods, Cluster Hypothesis. Clustering Algorithms.
2. File Structures, Inverted File, Suffix Trees & Suffix Arrays, Signature Files, Ring Structure, IR Models, Basic Concepts, Boolean Model, Vector Model, and Fuzzy Set Model. Search Strategies, Boolean Search, Serial Search, and Cluster based Retrieval, Matching Function
3. Performance Evaluation- Precision and Recall, Alternative Measures Reference Collection.
4. Taxonomy and Ontology: Creating Domain Specific Ontology, Ontology Life Cycle.
5. Multimedia IR Models & Languages- Data Modeling, Techniques to Represent Audio and Visual Document, Query Languages Indexing & Searching- Generic Multimedia Indexing Approach.
6. Searching the Web, Challenges, Characterizing the Web, Search Engines, Browsing.

Recommended Texts

1. Ricardo, B. Y., & Berthier, R. N. (2011). *Modern information retrieval: the concepts and technology behind search*. New Jersey, Boston: Addison-Wesley Professional.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Grossman, D. A., & Frieder, O. (2012). Information retrieval: Algorithms and heuristics (vol. 15). New York: Springer Science & Business Media.
2. Levene, M. (2011). An introduction to search engines and web navigation. Hoboken: John Wiley & Sons.
3. Subrahmanian, V. S., & Jajodia, S. (Eds.). (2012). Multimedia database systems: issues and research directions. New York: Springer Science & Business Media.
4. Djeraba, C. (Ed.). (2002). Multimedia mining: a highway to intelligent multimedia documents (vol. 22). New York: Springer Science & Business Media.
5. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge: Cambridge university press.

CSEC-7422 **Information Theory** **3(3+0)**

Information Theory is one of the few scientific fields fortunate enough to have an identifiable beginning - Claude Shannon's 1948 paper. The story of the evolution of how it progressed from a single theoretical paper to a broad field that has redefined our world is a fascinating one. It provides the opportunity to study the social, political, and technological interactions that have helped guide its development and define its trajectory, and gives us insight into how a new field evolves. The course will introduce students with concepts such as 1) Demonstrate knowledge and understanding of the fundamentals of information theory. 2) Appreciate the notion of fundamental limits in communication systems and more generally all systems. 3) Develop deeper understanding of communication systems. 4) Apply the concepts of information theory to various disciplines in information science, 5) understand entropy and entropy Ergodic theorem, 6) understand distortion and approximation techniques.

Contents

1. Information Sources
2. Pair Processes
3. Block Independent Channels, Conditionally Block Independent Channels, Stationarizing Block Independent Channels, Primitive Channels, Additive Noise Channels, Markov Channels
4. Finite-State Channels and Codes, Cascade Channels, Communication Systems, Couplings, Block to Sliding-Block: The Rohlin-Kakutani Theorem.
5. Entropy
6. The Entropy Ergodic Theorem
7. Distortion and Approximation
8. Approximating Random Vectors and Processes, The Monge/Kantorovich/Vasershtein Distance.
9. Variation and Distribution Distance, Coupling Discrete Spaces with the Hamming Distance, Process Distance and Approximation, Source Approximation and Codes, d-bar Continuous Channels.
10. Distortion and Entropy
11. Relative Entropy
12. Information Rates
13. Distortion and Information
14. Ergodic Theorems for Densities
15. Source Coding Theorems
16. Asynchronous Block Codes, Sliding-Block Source Codes, Geometric Interpretation, Properties of Good Source Codes, Optimal and Asymptotically Optimal Codes, Block Codes Sliding-Block Codes.

Recommended texts

1. Robert M. Gray. (2011). *Entropy and information theory,* (2nd ed.). New York: Springer.
2. John Scales Avery. (2012). *Information theory and evolution*, (2nd ed.). Bukit Batok: World Scientific Publishing Company.

Suggested Reading

1. Thomas M. Cover, Joy A. Thomas. (2006). *Elements of information theory*, (2nd ed.). Hoboken: Wiley-Interscience.
3. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7423 **Intelligent Multi Agent Systems** **3(3+0)**

A multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents. MAS can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may include methodic, functional, procedural, approaches, algorithmic, search or reinforcement learning. Intelligent MAS is an important area having multiple applications in Computing. They are a useful computational paradigm for creating systems that are flexible, adapt to change of the environment, and are able to integrate heterogeneous components. To address these characteristics, a number of issues are considered when studying MAS: how can an agent communicate and coordinate its activities with other agents in the system, how can agents represent and reason about the state of their interaction process, how can they represent and reason about actions, plans and knowledge of other agents, decompose goals and distribute tasks, what architecture can they be given so that they can solve a particular problem.

Contents

1. Intelligent Agents
2. Multi agent Systems
3. Agent Interaction Protocols
4. Distributed Problem Solving & Planning
5. Search Algorithm for Agents
6. Distributed Rational Decision Making
7. Learning in Multi agent Systems
8. Computational Organization Theory
9. Formal Methods in DAI (Distributed Artificial Intelligence)
10. Industrial and Practical Applications of DAI
11. Groupware and Computer Supported Cooperative Work
12. Distributed Models for Decision Support
13. Concurrent Programming for DAI

Recommended Texts

1. Gerhard Weiss. (2000). Multiagent systems: *A modern approach to distributed artificial intelligence* Cambridge: The MIT Press.
2. Tomas Salamon. (2011). *Design of agent-based models.* Brno: Eva & Tomas Bruckner.

Suggested Readings

1. Maria Fasli. (2007). *Agent technology for e-Commerce*, (1st ed.). Hoboken: Wiley.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7424 Intelligent Agents Technology 3(3+0)

The course focuses on the use of Intelligent Agent Technologies for development of complex software systems driven by collective intelligence. Intelligent software agents are such self-managed (autonomic) software entities that are capable to carry out some goal-driven and knowledge-based behavioral activities on behalf of a user or some other software application, which created it. This theory-oriented part of the course reviews appropriate AI methods and technologies needed to enable intelligent agents. After passing this course student will be able to understand about software agent technology including: What is agent technology? Technological contexts, Trends and drivers, Agent technologies and tools, Applications, Challenges and Technology roadmaps. The course focuses on the use of Distributed Artificial Intelligence methods, and more specifically of Intelligent Agents Technologies, for development of complex distributed software systems driven by collective intelligence. This theory-oriented part of the course reviews appropriate AI methods and technologies needed to enable intelligent agents.

Contents

1. Intelligent Agents: Attributes, Task level Skills, Knowledge, Communication Skills, End user Taxonomy of Agents, Types of Agents
2. Desktop Agents: Operating System Agents - An Intelligent Utility System Agent, Interface Agent, Application Agents, Email Agent, Search Agent, Application Suite Agents.
3. Internet Agents: Web search Agents, Benefits of Search Agents, Information Filtering Agents and its Benefits, Offline Delivery Agents and its Benefits, Notification Agents.
4. Intranet Agents: Intranet Search Agents, Informational Filtering Agents, Collaboration Agents, Lotus Notes Agents, Process Automation Agents – Edify work force.
5. Technology of Intelligent Agents: the historical view, technical view, Intelligence & Agency, Machinery, Content, Access & Security, Development Model, Steps towards Automation.
6. Agent Machinery: Principles of Reasoning – Inferencing Systems, Rules of Inferencing, Sensors & Actors, RAISE: An Inferencing System for Agents, The Trouble with Rules.

Recommended Texts

1. Salamon, T. (2011). *Design of agent-based models*. Brno: Eva & Tomas Bruckner Publishing.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Caglayan, A., & Harrison, C. (1997). *Agent sourcebook*. Hoboken: John Wiley & Sons, Inc.
2. Weiss, G. (Ed.). (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge: The MIT press.
3. Brenner, W., Zarnekow, R., & Wittig, H. (2012). *Intelligent software agents: foundations and applications*. New York: Springer Science & Business Media
4. Fasli, M. (2007). *Agent technology for e-commerce* (vol. 3). Hoboken: John Wiley & Sons.

CSEC-7425 **Machine Learning** **3(3+0)**

In this course, student will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work. More importantly, student will learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. The students will learn basic concepts and techniques of Machine Learning. They can develop skills of using recent machine learning software for solving practical problems. Students will gain experience of doing independent study and research. More specifically, it covers the following topics: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that a student could also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

Contents

1. Introduction to machine learning.
2. Linear regression.
3. Gradient Descent as general parameter Learning/optimization approach.
4. Logistic regression, One-vs-all classification, Regularization.
5. Neural Networks.
6. Practical advice for applying learning algorithms: How to develop, debugging, feature/model design, setting up experiment structure
7. Probability and Bayesian Analytics.
8. Naive Bayes.
9. Support Vector Machines (SVMs) and the intuition behind them.
10. Unsupervised learning: clustering and dimensionality reduction.
11. Anomaly detection.
12. Probabilistic Graphical Models: Bayesian Belief Networks.
13. Probabilistic Graphical Models: Learning from Data.
14. Probabilistic Graphical Models: EM algorithm.
15. Hidden Markov Models.

Recommended Texts

1. Christopher M. Bishop. (2006). *Pattern recognition and machine learning*. New York: Springer.
2. Tom Mitchell. (1997). *Machine learning*. New York: McGraw-Hill.

Suggested Readings

1. Trevor Hastie, Robert Tibshirani and Jerome Friedman. (2009). *The elements of statistical learning*. New York: Springer.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7426 Multimedia Systems Theory & Principles 3(3+0)

The rapid growth of multimedia technology has made it possible to deliver high quality audio, graphics, video and animation to the user. However, this growth in technology has not been met by a growth in design knowledge. While it is possible to have multimedia, it is not at all obvious that we know how to design high-quality multimedia systems that are fully usable to the degree we should expect. The course introduces students with theory and principles of multimedia contents constituting multimedia system. Different topics are given as a term paper. Extensive literature review is the part of this. Topics covered for term papers are Compression algorithms including Lossless compression, Image compression, Lossy compression, video compression techniques, audio compression techniques and MPEG compression. A prototype based on compression technique would be developed by students. Different Graphs and image representation schemes are also part of this course. Digital interfacing and video interfacing are also part of the course.

Contents

1. Introduction to Multimedia: What is Multimedia? Multimedia and Hypermedia, Overview of Multimedia Software Tools.
2. Graphics and Image Data Representations: Graphics/Image Data Types, Popular File Formats
3. Color in Image and Video: Color Science, Color Models in Images, Color Models in Video Fundamental Concepts in Video: Types of Video Signals, Analog Video, Digital Video.
4. Basics of Digital Audio: Digitization of Sound, MIDI: Musical Instrument Digital Interface, Quantization and Transmission of Audio.
5. Lossless Compression Algorithms: Introduction, Basics of Information Theory, Run-Length Coding, Variable-Length Coding (VLC), Dictionary-Based Coding, Arithmetic Coding, Lossless Image Compression.
6. Lossy Compression Algorithms: Distortion Measures, The Rate-Distortion Theory, Quantization, Transform Coding, Wavelet-Based Coding, Wavelet Packets.
7. Image Compression Standards: The JPEG Standard, The JPEG2000 Standard, The JPEG-LS Standard.
8. Basic Video Compression Techniques: Introduction to Video Compression, Video Compression Based on Motion Compensation, Search for Motion Vectors, H.261, H.263.
9. MPEG Video Coding MPEG-1 and 2.
10. MPEG Video Coding MPEG-4, 7.
11. Basic Audio Compression Techniques: ADPCM in Speech Coding, G.726 ADPCM, Vocoders.
12. MPEG Audio Compression: Psychoacoustics, MPEG Audio.

Recommended Texts

1. Z. M. Li and M. S. Drew. (2004). Fundamentals of multimedia. New York: Prentice Hall.
2. by N. Chapman and J. Chapman. (2004). Digital multimedia, (2nd ed.). Hoboken: Wiley.

Suggested Readings

1. Frank Y. Shih (2012) Multimedia security: watermarking, steganography, and forensics, (1st ed.). Boca Raton: CRC Press.
2. Daniel Cunliffe and Geoff Elliott (2005) Multimedia computing. Essex: Lexden Publishing Ltd

CSEC-7427 **Natural Language Processing** **3(3+0)**

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes, Context-free parsing, Earley algorithm, Extending CFG.
4. Probabilistic parsing, Parsing tricks, Human sentence processing.
5. Semantics, Forward-backward algorithm, Expectation Maximization.
6. Finite-state algebra, Finite-state implementation, Finite-state tagging, Noisy channels and FSTs, More FST examples.
7. Programming with regexps, Morphology and phonology.
8. Optimal paths in graphs, Structured prediction.
9. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Recommended Texts

1. 1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. Sebastopol: OReilly Media.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Bengio, Y. (2007). *Learning deep architectures for AI*.
4. Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge: MIT press

CSEC-7428 Parallel Processing: Algorithms and Architectures-I 3(3+0)

The course covers the needed perspective, taxonomy, and analysis tools for the understanding key concepts of parallel processing. It also delimits the models of parallel processing which has become quite important in recent years. More specifically, it deals with parallelism, its needs, type and its importance along with some simple architecture for better illustration, complexity of parallel algorithms (including asymptomatic, optimality and efficiency), PRAM algorithm along with its sub models and assumptions, selection and sorting networks and its design, Sorting and routing on a 2-D mesh, and other mesh related architectures. Paralleling processing covers Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory. Other Mesh-Related Architectures: Three or more dimensions, Stronger and weaker connectivities, meshes augmented with non-local links, meshes with dynamic links, Pyramid and multi-grid systems, Meshes of trees

Contents

1. Introduction to Parallelism: Why parallel processing? Parallel processing ups and downs, Types of parallelism: A taxonomy, Constraints of parallel processing, Effectiveness of parallel processing. Introduction to Parallel Algorithms: Some simple computations, some simple architectures, Algorithms for a linear array, Algorithms for a binary tree.
2. Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory.
3. PRAM and Basic Algorithms: PRAM submodels and assumptions, Data broadcasting, Semigroup or fan-in computation, Parallel prefix computation, Ranking the elements of a linked list, Matrix multiplication.
4. Sorting and Selection Networks: What is a sorting network? Figures of merit for sorting networks, Design of sorting networks, Batcher sorting networks
5. Sorting on a 2-D Mesh or Torus: Mesh-connected computers, Theshearsort algorithm.
6. Routing on a 2-D Mesh or Torus: Types of data routing, Useful elementary operations, Data routing on a 2-D array, Greedy routing algorithms.

Recommended Texts

1. Behrooz Parhami. (2006). *Introduction to parallel processing: algorithms and architectures*, New York: Springer.
2. Michel Cosnard and Denis Trystram (September 1996) *Parallel algorithms and architectures*, London: Intl Thomson Computer Pr (Sd).

Suggested Readings

1. Seyed H Roosta. (1999). *Parallel processing and parallel algorithms: theory and computation*, (1st ed.). New York: Springer.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7429 Parallel Processing: Algorithms and Architectures-II 3(3+0)

The course covers the needed perspective, taxonomy, and analysis tools for the understanding key concepts of parallel processing. It also delimits the models of parallel processing which has become quite important in recent years. More specifically, it deals with parallelism, its needs, type and its importance along with some simple architecture for better illustration, complexity of parallel algorithms (including asymptomatic, optimality and efficiency), PRAM algorithm along with its sub models and assumptions, selection and sorting networks and its design, Sorting and routing on a 2-D mesh, and other mesh related architectures. Paralleling processing covers Parallel Algorithm Complexity: Asymptotic complexity, Algorithm optimality and efficiency, Complexity classes, Parallelizable tasks and the NC class, Parallel programming paradigms, Solving recurrences. Models of Parallel Processing: Development of early models, SIMD versus MIMD architectures, Global versus distributed memory. Other Mesh-Related Architectures: Three or more dimensions, Stronger and weaker connectivities, meshes augmented with non-local links, meshes with dynamic links, Pyramid and multi-grid systems, Meshes of trees

Contents

1. Hypercubes and Their Algorithms: Definition and main properties, Embeddings and their usefulness, embedding of arrays and trees, A few simple algorithms, Matrix multiplication, Inverting a lower triangular matrix.
2. Sorting and Routing on Hypercubes: Defining the sorting problem, Bitonic sorting, Routing problems on a hypercube, Dimension-order routing, Broadcasting on a hypercube, Adaptive and fault-tolerant routing. Other Hypercubic Architectures: Modified and generalized hypercubes, Butterfly and permutation networks, Plus-or-minus-2i network, The cube-connected cycles, Shuffle and shuffle-exchange networks.
3. Sampler of Other Networks: Performance parameters for networks, Star and pancake networks, Ring-based networks, Composite or hybrid networks, Hierarchical (multi-level) networks
4. Data Storage, Input, and Output: Data access problems and caching, Cache coherence protocols
5. Reliable Parallel Processing: Defects, faults, failures, Defect-level methods, Fault-level methods, Error-level methods, Malfunction-level methods, Degradation-level methods

Recommended Texts

1. BehroozParhami. (2006). *Introduction to parallel processing*: *algorithms and architectures.* New York: Springer.
2. Michel Cosnard and Denis Trystram. (1996). *Parallel algorithms and architectures,* London: Intl Thomson Computer Pr (Sd).

Suggested Readings

1. Seyed H Roosta (December 10, 1999) *Parallel processing and parallel algorithms: theory and computation*, (1st ed.). New York: Springer.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7430 **Soft Computing** **3(3+0)**

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. It is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modeling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc. More specifically, it is collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low-cost solutions. It is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect, the role model for soft computing is the human mind.

Contents

1. Overview of Soft Computing,
2. Probabilistic Graphical Models,
3. Bayesian Networks,
4. Inferencing in Graphical Models,
5. Swarm Intelligence,
6. Ant Colony Optimization, Particle Swarm Optimization,
7. Cat Swarm Optimization,
8. Convergence Analysis,
9. Neural Networks,
10. Deep Learning,
11. Convolutional Neural Networks,
12. Recurrent Neural Networks
13. Relational Learning,
14. Tensor Factorization,
15. Ensemble Methods.

Recommended Texts

1. Engelbrecht, A. P. (2010, September). *Heterogeneous particle swarm optimization. In international conference on swarm intelligence* (pp. 191-202). New York: Springer.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT press.

Suggested Readings

1. Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* Cambridge: The MIT press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

CSEC-7431 **Speech Recognition and Synthesis** **3(3+0)**

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background:
2. N-gram Language Modeling.
3. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization J+M: Phonetics , J+M Speech Synthesis, J+M Word Classes and Part-of-Speech Tagging, but skip section 5.6 , Optional Advanced Reading: Chapter 4 "Text Segmentation and Organisation "Text Decoding" .
4. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software J+M, Read sections 1, 2, 3, 4, 5, and 6.1 and 6.1.1 from Alan Black's lecture notes on TTS and Festival. , Festival manual (used as Festival's scripting language): Introduction to Scheme for C Programmers, from Cal Tech. Optional Advanced Reading: "Prosody Prediction from Text" from Taylor, Paul. 2007.
5. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis) J+M New, pages 25-end Optional Advanced Reading:"Unit Selection Synthesis" from Taylor, Paul. 2007.
6. Optional Advanced Reading: Optional: Section 7 from Alan Black's lecture notes on TTS and Festival. Optional Advanced Reading: The rest of Section 6 of Alan Black's lecture notes
7. ASR: Noisy Channel Model, Bayes

Recommended Texts:

1. Jurafsky and Martin. (2014). Speech and language processing. (2nd ed.). Boston: Prentice Hall.

1. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Taylor, Paul. (2009). Text-to-speech synthesis. Cambridge: Cambridge University Press.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York:

Prentice Hall.

CSEC-7432 **Stochastic Processes** **3(3+0)**

The goal is to teach the theoretical concepts and techniques for solving problems that arises in practice. Beginning with the random variables, this course leads to the concept of stochastic process and linear filtering of random processes. More precisely, the objectives are 1. study of the basic concepts of the theory of stochastic processes; 2. introduction of the most important types of stochastic processes; 3. study of various properties and characteristics of processes; 4. study of the methods for describing and analyzing complex stochastic models. Various practical skills acquired during the study process would be: 1. understanding the most important types of stochastic processes (Poisson, Markov, Gaussian processes and others) and ability of finding the most appropriate process for modelling in particular situations arising in economics, engineering and other fields; 2. understanding the notions of ergodicity, stationarity, stochastic integration; application of these terms in context of financial mathematics.

Contents

1. Review of basic probability: Sample space, event, conditional probability, independent events, Baye's formula, random variable, distribution, cumulative distribution, Bernoulli distribution, binomial distribution, Poisson distribution, geometric distribution, density function, exponential distribution, gamma distribution, expectation, variance, standard deviation, joint
2. General Concepts: Definitions, Systems with Stochastic Inputs, The Power Spectrum, Discrete-Time Processes, Continuity, Differentiation, Integration, Shift Operators and Stationary Processes.
3. Finite Markov chains: basics, examples, 1-step and n-step transition probabilities, stationary distributions, classifying states, periodicity of classes, absorption of transient states, reversibility
4. Branching processes: probability generating function, compound distribution
5. Renewal theory: sequence generating function, pattern generation, consecutive successes, mean number of trials, breaking even, mean number of occurrences, comparison of patterns
6. Markov processes: Poisson processes, extensions, pure birth processes, Yule process, pure death processes, birth and death processes, linear case, linear growth with immigration, M/M/infinity queue, power supply problem, stationary distributions, examples, absorption
7. Brownian motion: definition, no drift case, reaching a point, avoiding zero, returning to zero, drift.
8. Random Walks and Other Applications: Random Walks, Cyclostationary Processes
9. The Schwarz Inequality.
10. Mean Square Estimation: Introduction, Prediction.

Recommended Texts

1. Bhattacharya, R. N., & Waymire, E. C. (2009). Stochastic processes with applications (vol. 61). Philadelphia: SIAM.

Suggested Readings

1. Brzezniak, Z., & Zastawniak, T. (2000). Basic stochastic processes: a course through exercises. New York: Springer Science & Business Media.
2. Nakagawa, T. (2011). Stochastic processes: With applications to reliability theory. New York: Springer Science & Business Media.
3. Papoulis, A., & Pillai, S. U. (2002). Probability, random variables, and stochastic processes. New

York: Tata McGraw-Hill Education.

CSEC-7433 Stochastic Systems 3(3+0)

Exploration of noise, uncertainty, and randomness in the context of signals and systems. The course will introduce discrete- and continuous-time random processes as input and/or output signals of various types of systems, with and without memory or feedback. Probabilistic notions will be tightly integrated with techniques from signals and systems, such as linearity, time-invariance, causality, transform methods, and stability. Basic concepts will be illustrated via numerous examples, such as noise in linear and nonlinear circuits, average consensus and PageRank, queuing systems, noise in remote sensing applications, Bayesian filtering, Monte Carlo simulation, risk allocation in financial portfolios, stochastic gradient descent. Upon successful completion of the course, students will be able to reason about noise, mean and variance, joint moments, sequences of random variables, stochastic convergence and limit theorems, uncertainty, and randomness in the context of engineering systems using tools and techniques from probability theory and systems theory. Student will be able to understand, analyze, and solve typical problems in statistics.

Contents

1. Overview of Probability: Introduction, Definitions, Probability and Induction.
2. Causality Versus Randomness.
3. The Axioms of Probability: Set Theory, Probability Space, Conditional Probability.
4. Repeated Trials: Combined Experiments.
5. Bernoulli Trials, Bernoulli's Theorem and Games of Chance.
6. The Concept of a Random Variable: Introduction, Distribution and Density Functions, Specific Random Variables,
7. Conditional Distributions, Asymptotic Approximations for Binomial Random Variable.
8. Functions of One Random Variable: The Random Variable g(x), The Distribution of g(x).
9. Mean and Variance.
10. Moments, Characteristic Functions.
11. Two Random Variables: Bivariate Distributions, One Function of Two Random Variables, Two Functions of Two Random Variables.
12. Joint Moments, Joint Characteristic Functions, Conditional Distributions, Conditional Expected Values.
13. Sequences of Random Variables: General Concepts.
14. Conditional Densities, Characteristic Functions, and Normality.
15. Mean Square Estimation.
16. Stochastic Convergence and Limit Theorems.
17. Random Numbers: Meaning and Generation.
18. Statistics: Introduction, Estimation.
19. Parameter Estimation, Hypothesis Testing.

Recommended Texts

1. Klyatskin, V. I. (2005). *Dynamics of stochastic systems*, (1st ed.). Amsterdam: Elsevier.
2. Özsu, M. T., & Valduriez, P. (1999). *Principles of distributed database systems* (vol. 2). New York: Prentice Hall.

Suggested Readings

1. Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. New York: Tata McGraw-Hill Education.
2. Adomian, G. (Ed.). (2014). *Applied stochastic processes*. Cambridge: Academic press.

CSEC-7434 **Advanced Theory of Computation** **3(3+0)**

The course aims to introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability. Several mathematical models of computation have been formulated independently and under any such computational model, the existence of well-defined but unsolvable problems can be formally shown. These topics form part of the core of the mathematical foundations of computer science that will provide students and researchers with a sound theoretical view of the most fundamental concepts of computation. Specifically, this course provides a rigorous introduction to the theoretical foundations of computer science. It deals with a number of interconnected topics and tries to answer the basic questions, "What is a computer?", "What is an algorithm?", and "What is computable?". This course examines important theorems and proofs, establishes a number of interesting assertions in order to expose the techniques used in the area of theory of computation. Note that although this is not a "mathematics" course, it does make significant use of mathematical structures, abstractions, definitions, theorems, proofs, lemmas, corollaries, logical reasoning, inductive proofs, and the like. If such concepts are difficult for you, you will find this course very difficult but rewarding.

Contents

1. Mathematical Tools and Techniques: Logic and Proofs, Sets, Functions and Equivalence Relations, Languages.
2. Proofs, Principle of Mathematical Induction, Recursive Definitions, Structural Induction.
3. Regular Languages and Regular Expression.
4. Nondeterministic Finite Automata with A-Transitions, Kleene's Theorem.
5. Regular, Non-regular Languages, Criteria for Regularity, Minimal Finite Automata
6. Context-Free Grammars, Regular Grammars, Derivation Tree and Ambiguity
7. Pushdown Automata: Definitions and Examples, Deterministic Pushdown Automata
8. Context-Free, Non-Context-Free Languages.
9. Turing Machines: Definitions and Examples, Non-deterministic TM, Universal TM
10. Recursively Enumerable and Languages: Recursively Enumerable and Recursive, Enumerating a Language, More General Grammars, Context-Sensitive Languages and the Chomsky Hierarchy
11. Unsolvable Problems: A No-recursive Language and an Unsolvable Problem, The Halting Problem, Unsolvable Problems Involving TMs, Rice's Theorem
12. Computable Functions: Primitive Recursive Functions, PRF and Bounded Operations, Unbounded. Minimalization and ų-Recursive Functions, Godel Numbering

Recommended Texts

1. John Martin. (2010). *Introduction to languages and the theory of computation*, (4th ed). New York: McGraw-Hill Science/Engineering/Math.
2. Tourlakis, George J. (2012). *Theory of computation*. Hoboken: Wiley.

Suggested Readings

1. Goddard, W. (2008). Introducing the theory of computation. **Burlington: Jones and Bartlett Publishers, Inc.**
2. Dexter C. Kozen. (2010). Theory of computation, (1st ed). New York: Springer.

CSEC-7435 **Advanced Operating Systems** **3(3+0)**

Advanced Operating Systems course addresses a broad range of topics in operating system design and implementation, including: Operating system structuring, Synchronization, communication and scheduling in parallel systems, Distributed systems, their communication mechanisms, distributed objects and middleware, Failures and recovery management, CPU scheduling: Scheduling policies and algorithms, Scheduling algorithm comparison, Real-time and multi-processor scheduling, Linux case study. Dynamic memory management: Internal design alternatives for malloc and free routines, Garbage collection. OS memory management: Memory protection, Program relocation, Memory partitioning techniques, Virtual memory, Paging and segmentation, TLB and cache management. File systems: Naming issues, Design alternatives for file systems, Example file systems and their comparison. I/O management: Memory mapped Vs Direct I/O, Interrupt driven Vs Polled I/O, Device controllers and device drivers, Naming issues, OS I/O architecture, Buffering techniques, Disk devices and their management. At the end of the course, students are expected to be proficient in details of operating systems and be sensitive to implementation and performance tuning of operating systems
in preparation to entering the industry or in pursuit of graduate studies.

Contents

1. Introduction: Operating System Architecture, Multitasking, Multiuser, Multiprocessing, Multi-Threading OS, Operating System Services for Process Management.
2. Process Scheduling Concepts, System Calls for Process Management, Process Communication and Synchronization Concepts, Memory and I/O Management Overview, UNIX Commands for System Administration.
3. Multitasking OS: Design and Implementation Kernel of Multitasking OS: Services, Process State Transitions, Functional Specification, Implementation Considerations, System List.
4. Multiprocessor Systems: Introduction, Parallel Hardware and Interconnections, Types of Multiprocessor OS, Sharaing OS, Mutiprocesor OS Design Considerations, Threads, Thread Scheduling, Kernel Mode Processes, Multiprocessor Synchronization.
5. Memory Management: Overview, Pages, Zones, Kmalloc, Vmalloc, Slab Layer, Slab Layer Allocator, Deallocator, Statically Allocating on The Stack, High Memory Mapping. Non-Contiguous Memory.
6. Unit V I/O Systems: I/O Device Types, I/O Structure, Driver Interfaces, Disk Device Driver Access Strategies, Unification of Files And I/O Devices, Generalized Disk Device Drivers, Disk Caching
7. File Descriptors, File Blocks Allocaton, Mapping of File Blocks, System Calls for The File System: Open, Read, Write, Lseek, Close. Mounting and Un Mounting File Systems, Link, Unlink,

Recommended Texts

1. Stallings, W. (2012). *Operating systems: internals and design principles*. Boston: Prentice Hall.
2. Tanenbaum, A. S., & Woodhull, A. S. (2006). *Operating systems design and implementation*, (3rd ed.). Boston: Prentice Hall.

Suggested Readings

1. Maurice J. Bach. (1986). *The design of the UNIX Operating System*, Boston: Prentice Hall.
2. Andrew S. Tanenbaum. (2007). *Modern operating systems*, (3rd ed). Boston: Prentice Hall.
3. Stevens, W. R., Rudoff, A. M., & Fenner, B. (2003). *UNIX network programming volume 1: The Sockets Networking API* (vol. 3). Boston: Addison-Wesley Professional.
4. Love, R. (2010). *Linux kernel development*. New York: Pearson Education.

CSEC-7436 Swarm Intelligence 3(3+0)

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. The course is aimed to provide students' knowledge of swam principles, experiments and models. They will learn about mobile robotics, machine-learning techniques, Multi-level modeling of self-organized robotic systems, combined modeling and machine-learning methods for control optimization, aggregation and segregation and Distributed structure building in natural and artificial systems. More specifically students would study to evaluate the power and limitation of Swarm Intelligence when it comes to solving real life problems, to advance the state of the art in Swarm Intelligence, and to introduce students to academic research. There will be a course project in which teams of students apply swarm intelligence to a problem of their choosing. This application should advance the state of knowledge in swarm intelligence. You will be asked to research, design and build a swarm system. Course will also include few practical exercises.

Contents

1. Introduction to swarm intelligence and key principles (e.g., self-organization, stigmergy), natural and artificial examples, computational and embedded SI. Foraging, trail laying/following mechanisms. Open-space, multi-source foraging experiments: biological data and microscopic models.
2. From real to virtual ants
3. Introduction to mobile robotics
4. Introduction to unsupervised multi-agent machine-learning techniques for automatic design and optimization: terminology and classification, Genetic Algorithms (GA) and Particle Swarm optimization (PSO). Comparison between both techniques in theory and practice.
5. Application of machine-learning techniques to automatic control design and optimization in single-robot and multi-robot experiments. Specific issues for automatic control design and optimization in collective systems
6. Collective movements in natural societies; focus on flocking phenomena. Collective movements in artificial systems: Reynolds' virtual agents (Boids) and experiments with multi-robot systems
7. Multi-level modeling of self-organized robotic systems: microscopic and macroscopic models; Markov formalism; linear and nonlinear micro-to-macro mapping, model analysis.
8. Combined modeling and machine-learning methods for control optimization. Diversity and specialization metrics.
9. Division of labor and task-allocation mechanisms, part I: threshold-based algorithms.

.

Recommended Texts

1. E. Bonabeau, M. Dorigo, and G. Theraulaz. (1999). Swarm intelligence: from natural to artificial systems, Oxford: Oxford University Press.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

Suggested Readings

5. Maurice J. Bach. (1986). *The design of the UNIX Operating System*, Boston: Prentice Hall.

6. Andrew S. Tanenbaum. (2007). *Modern operating systems*, (3rd ed). Boston: Prentice Hall.

CSEC-7437 Evolutionary Computation 3(3+0)

It intended as a fundamental introduction, as well as a survey of the many aspects of evolutionary algorithms (EAs), in particular GA, GP, ES, and will concentrate on the basic concepts of representation, operators and overall control, followed by examples of the use of these concepts in important applications. As such, this course will not do much in-depth study of any one area. Rather the course is intended as a good introduction for those who have had no exposure to EC and as a stepping stone for those interested in more specific areas. At the end of this course, student would be able to: Explain evolutionary computation techniques and methodologies set in the context of modern heuristic methods. , apply various evolutionary computation methods and algorithms for particular classes of problems, develop evolutionary algorithms for real-world applications, and use scientific research papers and present them in a seminar.

Contents

1. Introduction to Evolutionary Computation, Biological and artificial evolution, Evolutionary computation and AI, Different historical branches of EC, e.g., GAs, EP, ES, GP, A simple evolutionary algorithm.
2. Search Operators, Recombination/Crossover for strings (e.g., binary strings), e.g., one-point, multi-point, and uniform crossover operators, Mutation for strings, e.g., bit-flipping, Recombination/Crossover and mutation rates, Recombination for real-valued representations, e.g., discrete and intermediate recombination, Mutation for real-valued representations, e.g., Gaussian and Cauchy mutations, self-adaptive mutations, etc, Why and how a recombination or mutation operator works?
3. Selection Schemes, Fitness proportional selection and fitness scaling, Ranking, including linear, power, exponential and other ranking methods, Tournament selection, Selection pressure and its impact on evolutionary search
4. Search Operators and Representations, Mixing different search operators, An anomaly of self-adaptive mutations, The importance of representation, e.g., binary vs. Gray coding
5. Evolutionary Combinatorial Optimization, Evolutionary algorithms for TSPs, Evolutionary algorithms for lecture room assignment, Hybrid evolutionary and local search algorithms
6. Co-evolution, Cooperative co-evolution, Competitive co-evolution
7. Niching and Speciation, Fitness sharing (explicit and implicit), Crowding and mating restriction
8. Constraint Handling, Common techniques, e.g., penalty methods, repair methods

Recommended Texts

1. Andries Engelbrecht (2007) *Computational intelligence: An introduction.* Hoboken: Wiley
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

Suggested Readings

1. Z. Michalewicz and D. B. Fogel, (2000) *How to solve it: Modern heuristics*, New York: Springer.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

CSEC-7438 **Fog Computing** **3(3+0)**

Fog computing has emerged because cloud computing hasn't been able to keep up with the growth of IoT devices. The increase of consumer and enterprise devices connected to the IoT has put too much strain on cloud services from even the most cutting-edge providers. The reason is that the cloud is too far away from the point of origin, and sending data to the cloud to be analyzed results in latency that is simply unacceptable in many environments. This course is designed to give a multitude of technologies that comprise the modern concept of Fog and Edge computing, their applications, architectures, and technologies. Fog Computing is an emergent architecture for computing, storage, control, and networking that distributes these services closer to end users along the cloud-to-things continuum. It covers both mobile and wire line scenarios, traverses across hardware and software, resides on network edge but also over access networks and among end users, and includes both data plane and control plane.

Contents

1. Overview: From Cloud to Fog, Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers
2. Overview: From IT to loT to IoE, Basic attributes and challenges for IoE
3. Fog/Edge Computing ComputingArchitectures, Feasible Service Models of Fog
4. Fog/Edge Computing Applications: Smart city, Smart health, Smart Building, Internet of Energy, smart transportation
5. Fog/Edge/IoT Data storage and Management, Real-time streaming data processing
6. Resource-constrained devices management and optimization
7. Scheduling for Fog/Edge infrastructures
8. Modeling and Simulation of Fog and Edge Computing Environments using iFogSim Toolkit:
9. Components of iFogsim: physical components and logical components
10. Installation of iFogSim, Building Simulation with iFogSim
11. Application Case Studies: a latency-sensitive online game, Intelligent Surveillance through Distributed Camera Networks, Performance Evaluation
12. Example Scenarios: Sensors with different tuple emission rate, send specific number of tuples from a sensor, connect lower level Fog devices with nearby gateways, Mobility of a Fog device, Make Cluster of Fog devices, Simulation of a Placement Policy
13. Common Standards in Fog Computing: The Open Fog Consortium, Standards of Application
14. Building and simulating a Use-Case Study in Smart Health / Smart Cities / Smart Government / Smart Industrial Environments / Industry 4.0
15. Fog/Edge/IoT Reliability, Security and Privacy, Fog/Edge privacy challenges and solutions

Recommended Texts

1. Mahmood Zaigham. (2018). *Fog computing concepts, frameworks and technologies*

Suggested Readings

1. Rajkumar Buyya, Satish Srirama. (January, 2019). *Fog and edge computing: principles and*

paradigms, Hoboken: Wiley Press.

CSEC-7439 Advanced Computer Architecture 3(3+0)

To develop an understanding of high-performance computer architecture, as a foundation for advanced work in computer architecture. The students will get overview of computer architecture, which stresses the underlying design principles and the impact of these principles on computer performance. More specifically, Distinction between Computer Architecture, Organization and design, General operation of a stored program digital computer, The Fetch – Execute process, Concept of an ISA, Instruction set features, Addressing Modes, RISC and CISC architectures, Measures of performance, Implementing Register Transfers using Digital Logic Circuits, The Design Process, A Uni-Bus implementation for the SRC, Pipelining, Microprogramming. General topics include design methodology, processor design, control design, memory organization, system organization, and parallel processing. At the end, a student would be able to know the classes of computers, and new trends and developments in computer architecture, understand pipelining, instruction set architectures, memory addressing, understand the performance metrics of microprocessors, memory, networks, and disks.

Contents

1. Introduction to Advanced Computer: Flynn's Taxonomy of Computer Architecture, SIMD, MIMD, Interconnection Networks.
2. Multiprocessors Interconnection Networks: Interconnection Networks Taxonomy, Bus-Based Dynamic Interconnection Networks.
3. Performance Analysis of Multiprocessor Architecture: Computational Models, An Argument for Parallel Architectures, Interconnection Networks Performance Issues.
4. Shared Memory Architecture: Classification of Shared Memory Systems, Bus-Based Symmetric Multiprocessors.
5. Message Passing Architecture: Introduction to Message Passing, Routing in Message Passing Networks, Switching Mechanisms in Message Passing.
6. Abstract Models: The PRAM Model and Its Variations, Simulating Multiple Accesses on an EREW PRAM, Analysis of Parallel Algorithms.
7. Network Computing: Computer Networks Basics, Client/Server Systems, Clusters, Interconnection Networks, Cluster Examples, Grid Computing.
8. Parallel Programming in the Parallel Virtual Machine: PVM Environment and Application Structure, Task Creation, Task Groups.
9. Message Passing Interface (MPI): Communicators, Virtual Topologies, Task Communication, Synchronization, Collective Operations, Task Creation.
10. Scheduling and Task Allocation: The Scheduling Problem, Scheduling DAGs without Considering, Communication, Communication Models.

Recommended Texts

1. El-Rewini, H., & Abd-El-Barr, M. (2005). Advanced computer architecture and parallel processing (Vol. 42). Hoboken: John Wiley & Sons.

Suggested Readings

1. Shiva, S. G. (2018). Advanced computer architectures. Boca Raton: CRC Press.
2. Amit Kumar Mishra, S K Kataria & Sons. (2010). Advanced Computer Architecture, New Delhi: S K Kataria & Sons.
3. DezsoSima. (1997). Advanced computer architecture: A design space approach, Boston: Addison-

Wesley.
4. Hwang, K., & Xu, Z. (1998). Scalable parallel computing: technology, architecture, programming. New York: McGraw-Hill, Inc.

CSEC-7440 **Advance Natural Language Engineering** **3(3+0)**

In this course we will explore statistical, model-based approaches to natural language processing. There will be a focus on corpus-driven methods that make use of supervised and unsupervised machine learning approaches and algorithms. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. In this course, students will gain a basic introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Text Classification via Naive Bayes & Maximum Entropy, Classification Tutorial, MaxEnt Tutorial, Generative and Discriminative Classifiers.
2. Hidden Markov Models, Clustering, Part-of-Speech Tagging, TnT Tagger.
3. Advanced Part-of-Speech Tagging, Word Alignments, MT Tutorial, Overview, IBM Models, HMM-Alignments, Agreement.
4. Phrase-Based Translation, Decoding, Phrases
5. Syntactic Parsing, Best-First, A*, K-Best, Shift-Reduce Parsing.
6. Advanced Constituency Parsing, Unlexicalized, Lexicalized, Latent Variable.
7. Semantic Parsing, Hierarchical (Syntax-Based) Translation, Hiero, GHKM, Syntax vs. Phrases, Synchronous Grammars.
8. Sentiment Analysis, Aspects, Lexicons, Summarization.
9. Summarization, Query, N-Gram, Topical.
10. Lexical Acquisition.

Recommended Texts

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: The MIT press.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. Boston: OReilly Media. *Reading, Massachusetts*.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge: Cambridge University Press.
3. Bengio, Y. (2007). Learning deep architectures for AI (Technical Report 1312). *Université de Montréal, dept. IRO*. Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge: The MIT press.

CSEC-7441 Special Topics in Natural Language Engineering 3(3+0)

This course covers a broad range of topics in computational linguistics/natural language processing, including word and sentence tokenization, text classification and sentiment analysis, spelling correction, information extraction, parsing, meaning extraction, and question answering, We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. In recent years, deep learning (or neural network) approaches have obtained very high performance across many different NLP tasks, using single end-to-end neural models that do not require traditional, task-specific feature engineering. In this course, students will gain a thorough introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Logistic Regression: Classification: The sigmoid; Learning in Logistics Regression.
2. The Cross-Entropy loss function; Gradient Descent, Regularization, etc.
3. Vector Semantics: Lexical Semantics, Vector Semantics, Words and Vectors.
4. Cosine for measuring similarity, TF-IDF Weighting terms in the vector, etc.
5. Neural Networks and Neural Language Models.
6. Units, The XOR problem, Feed Forward Neural Network, Training Neural Nets, etc.
7. Part-of-Speech Tagging.
8. Word Classes, Penn POS Tagset, POS Tagging, HMM POS Tagging, etc.
9. Sequence Processing with Recurrent Networks Simple recurrent neural network, Applications of Recurrent Neural Networks.
10. Deep Networks: Stacked and Bidirectional RNNs, Managing Context in RNNs: LSTMs and GRUs, Words, Subwords and Characters.
11. Encoder-Decoder Models.
12. Attention, and Contextual Embeddings

Recommended Texts

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: The MIT press.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. Boston: OReilly Media. *Reading, Massachusetts*.
2. Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

CSEC-7442 **Text Processing and Data Analysis** **3(3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing, Text Processing Functions.
6. Loop Control, Hashes
7. DBM Databases, Advanced Sorting.
8. Regular Expressions
9. Environment Variables, CGI-Programming.
10. Process Management, References and Data Structures.
11. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. Sebastopol: O'Reilly & Associates, Inc.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming Perl: Unmatched power for text processing and scripting*. Sebastopol: O'Reilly Media, Inc.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl cookbook: solutions & examples for perl programmers*. Sebastopol: O'Reilly Media, Inc.
2. Lidie, S., & Walsh, N. (2002). *Mastering Perl/Tk: graphical user Interfaces in perl*. Sebastopol: O'Reilly Media, Inc.

CSEC-7443 Speech Recognition & Synthesis 3(3+0)

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background: N-gram Language Modeling
2. TTS: Background (part of speech tagging
3. Machine learning, classification, NLP) and Text Normalization
4. Phonetics Speech Synthesis, pages 1-10 , Optional Advanced Reading, Text Segmentation and Organisation, Text Decoding.
5. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software, Prosody Prediction from Text.
6. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis), Unit Selection Synthesis, Optional Advanced Reading.
7. ASR: Noisy Channel Model, Bayes
8. HMMs, Forward, Viterbi, Hidden Markov Models, Automatic Speech Recognition.
9. ASR: Feature Extraction and Acoustic Modeling
10. Evaluation, Speech Recognition: Advanced Topics.
11. ASR: Learning (Baum-Welch) and Disfluencies, Automatic Speech Recognition, Speech Recognition: Advanced Topics.

Recommended Texts

1. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

Suggested Readings

2. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: The MIT press.
3. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

CSEC-7444 **Advanced Algorithm Analysis** **3(3+0)**

This is an advanced course in design and analysis of algorithms covering topics typically not covered in undergraduate algorithms. It includes introduction of formal techniques and the underlying mathematical theory. NP-completeness. Search Techniques. Randomized Algorithms. Heuristic and Approximation Algorithms. Topics include asymptotic analysis of upper and average complexity bounds using big-O, little-o, and theta notation. Fundamental algorithmic strategies (brute-force, greedy, divide-and-conquer, backtracking, branch-and-bound, pattern matching, and numerical approximations) are covered string matching, parallel programming. Also, include are standard graph and tree algorithms. Additional topics include standard complexity classes, time and space tradeoffs in algorithms, using recurrence relations to analyze recursive algorithms, non-computable functions, the halting problem, and the implications of non-computability. Algorithmic animation is used to reinforce theoretical results. Upon completion of the course, students should be able to explain the mathematical concepts used in describing the complexity of an algorithm, and select and apply algorithms appropriate to a particular situation.

Contents

1. Dynamic Programming, elements, rod cutting, Longest Common Subsequence, Optimal Binary Search Trees.
2. Greedy Algorithms and methods.
3. Amortized Analysis: Aggregate Analysis, Accounting Method, Potential Method, Dynamic Tables.
4. Fibonacci Heaps: Structure of Fibonacci Heaps, Mergeable-Heap Operations.
5. VanEmde Boas Trees: Preliminary Approaches, A Recursive Structure, The Van Emde Boas Tree.
6. Dijkstra algorithm, bellmen ford algorithm and proof of shortest paths.
7. All-Pairs Shortest Paths: Shortest Paths and Matrix Multiplication, The Floyd-Warshall Algorithm, Johnson's Algorithm for Sparse Graphs.
8. Maximum Flow: Flow Networks, The Ford-Fulkerson Method, Maximum Bipartite Matching, Push-Relabel Algorithms, The Relabel-To-Front Algorithm.
9. Multithreaded Algorithms: The Basics of Dynamic Multithreading
10. String Matching: The Naive String-Matching Algorithm
11. Approximation Algorithms: The Vertex-Cover Problem, The Traveling-Salesman Problem, The Set-Covering Problem, Randomization and Linear Programming, The Subset-Sum Problem.
12. Parallel Algorithms: Parallelism, The PRAM Model, Simple Parallel Operations.

Recommended Texts

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Cambridge: MIT press.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. McConnell, J. J. (2001) *Analysis of algorithms: An active learning approach*, (2nd ed.). Burlington: Jones and Bartlett Publishers.

2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

SEEC-7401 Reliability Engineering 3 (3+0)

One of the most difficult issues in developing a system product is determining how long it will last and how long it should last. Reliability describes the ability of a system or component to function under stated conditions for a specified period of time. Reliability is closely related to availability, which is typically described as the ability of a component or system to function at a specified moment or interval of time. This idea of reliability that was birthed in the early days of electronics and aviation, now extends into every sector of consumer and industrial products. Automobiles, airplane, televisions and computers have all found their way into the hands of everyday consumers because of the advancement in reliability engineering. The main objectives of reliability engineering are: To apply engineering knowledge and specialist techniques to prevent or to reduce the likelihood or frequency of failures. To identify and correct the causes of failures that occur despite the efforts to prevent them.

Contents

1. Introduction and Overview, LFG Basics.
2. LFG Basics II, Templates I, MacOSX, Unix.
3. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file.
4. Lexical Rules, Passive and Argument alternations.
5. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic
6. Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation.
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.
11. Finite-State Morphology (FSM) I.
12. FSM II (-unknown), Free Word Order and the Shuffle Operator.
13. Meta-cateogries, Metarulemacros and Coordination.
14. Project

Recommended Texts

1. Ebeling, C. E. (2009). *An introduction to reliability and maintainability engineering*, (2^{nd} ed.). Illinois: Waveland Press, Inc.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

Suggested Readings

1. Michael R. Lyu. (2008). *IEEE recommended practice in software reliability handbook of software reliability engineering,* New York: Published by IEEE Computer Society Press and McGraw-Hill Book Company.
2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

SEEC-7402 **Research Methodology** **3(3+0)**

Research can be defined as "an activity that involves finding out, in a more or less systematic way, things you did not know. Methodology is the philosophical framework within which the research is conducted or the foundation upon which the research is based. Students will understa-nd research terminology, research objectives and importance. They will become aware of the ethical principles of research, ethical challenges, approval processes and types of researches. It will provide an opportunity for participants to establish or advance their understanding of research through critical exploration of research language, ethics, and approaches. The course introduces the language of research, ethical principles and challenges, and the elements of the research process within quantitative, qualitative, and mixed methods approaches. Participants will use these theoretical underpinnings to begin to critically review literature relevant to their field or interests and determine how research findings are useful in forming their understanding of their work, social, local and global environment.

Contents

1. Introduction to Research. Objectives of Research. Importance of Research Methodology in Research Study.
2. Types of Research. Steps in Conducting Research.
3. What is Literature Review? Why need for Literature Review. Types of Literature Review. Systematic Literature Review Protocol.
4. Problem Statement and Problem formulation. Criteria for selecting a problem.
5. Identifying Types of variables in Research.
6. Types of hypothesis. Identifying Target Population. Types of Sampling. Sampling Techniques.
7. Quantitative Research Methods.
8. Scientific Methods. Design of Quantitative Surveys. Techniques to Conduct Quantitative Methods.
9. Introduction to Qualitative Research. Qualitative Research Methods.
10. Data Analysis and Theory in Qualitative Research Articles.
11. Introduction to Mixed Methods Research. Design of Mixed Methods Research. Evaluation of Mixed Methods Research.
12. Case Study. How to Conduct a Case Study. Case Study Protocol.

Recommended Texts:

1. Creswell, J. W. (2014*). Research design: Qualitative, quantitative and mixed methods approaches*, (2nd ed.). Thousand Oaks: SAGE.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

3. Michael R. Lyu. (2008). *IEEE recommended practice in software reliability handbook of software reliability engineering,* New York: Published by IEEE Computer Society Press and McGraw-Hill Book Company.
4. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

SEEC-7403 **Software Configuration Management** **3(3+0)**

This course provides a basic introduction to the theory, principles, and techniques of Software Configuration Management (SCM) as it applies to the entire software lifecycle. It addresses the application of SCM in a wide variety of approaches to software development and maintenance, from traditional to agile. The course illustrates the SCM strategies, techniques, and required tool capabilities that support each of the activities in the software development life cycle. The student will also gain a value-based understanding of which SCM techniques are most useful for the development approach and tool capabilities that currently exist in their company. The course concludes by providing an SCM implementation framework for planning and selecting the optimal SCM strategy and tool capabilities for both the project and organizational levels. More specifically, it deals with Management of the SCM Process, Constraints and Guidance for the SCM Process, Surveillance of SCM, Software Configuration Identification, SCM tools and current research in SCM.

Contents

1. Management of the SCM Process
2. Organizational Context for SCM
3. Constraints and Guidance for the SCM Process
4. Planning for SCM. SCM Plan
5. Surveillance of Software Configuration Management
6. Software Configuration Identification
7. Identifying Items to Be Controlled
8. Software Library
9. Software Configuration Control
10. Requesting, Evaluating, and Approving Software Changes
11. Implementing Software Changes
12. Deviations and Waivers
13. Software Configuration Status Accounting
14. Software Configuration Status Information
15. Software Configuration Status Reporting. Software Configuration Auditing
16. Software Functional Configuration Audit. Software Physical Configuration Audit
17. In-process Audits of a Software Baseline

Recommended Texts

1. Tephen P. Berczuk, Brad Appleton. (2003). *Software configuration management patterns: effective teamwork, practical integration*. Boston: Addison Wesley.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. Michael R. Lyu. (2008). *IEEE recommended practice in software reliability handbook of software reliability engineering,* New York: Published by IEEE Computer Society Press and McGraw-Hill Book Company.

2. Andries Engelbrecht, (2007) Computational intelligence: an introduction. Hoboken: Wiley.

SEEC-7404 Software Measurement and Metrics 3(3+0)

A measurement is a manifestation of the size, quantity, amount or dimension of a particular attributes of a product or process. Software measurement is a titrate impute of a characteristic of a software product or the software process. It is an authority within software engineering. Software is measured to create the quality of the current product or process, anticipate future qualities of the product or process, enhance the quality of a product or process and regulate the state of the project in relation to budget and schedule. Measurement is critical for the successful management of software projects. It is the basis of project planning because it is used to establish achievable project targets. Measurement helps to monitor the progress of a project. A project is considered successful if it has met its targets in terms of the cost, the schedule, and the quality. It is possible to evaluate these by measuring the project-related data. The overall objective of this course is to determine adjusted function point count for a software system.

Contents

1. Introduction to quality control and planning needs
2. Measurement Concepts
3. Measurement as a support process
4. Review Metrics Models and Standards
5. Measurement goals
6. Formulating problem and goal statement
7. Prioritize information needs and objectives
8. Formalize measurement goals
9. Specify Measures
10. Identify questions and indicators
11. Identify data elements, Operational definitions for measures.
12. Specify Data Collection and Storage Procedures.
13. Sources of data. How to collect and store the measurement data?

Recommended Texts

1. Stephen H. Kan. (2003). Metrics and models in software quality engineering. Boston: Addison Wesley.
2. Chris Chen and Hadley Roth. (2005). The big book of six sigma training games. New York: McGraw-Hill.

Suggested Readings

1. Anita Carleton, William A. Flora. (1999). *Measuring the software process.* Boston: Addison-Wesley.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

SEEC-7405 **Software Risk Management** **3(3+0)**

Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. A possibility of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor-quality software, not being able to complete the project on time. This course focuses on intermediate and advanced strategies you can use to manage general risks and details practical techniques you can use to control your project's specific risks. More specifically, this course is to introduce you to risk management, risk management paradigm, risk identification and its management in CMM, risk taxonomy, various tools for identifying the risks, classifying risks, its analysis, planning, and monitoring, best practices and lessons learned on Risks, various other Case study based exercises and presentations on approach by each team.

Contents

1. What is risk and risk management?
2. Motivation for risk management.
3. Reasons we don't do risk management.
4. SEI's Risk Management paradigm.
5. Identifying and recording software risk.
6. Risk Taxonomy.
7. Tools and methods for identifying and recording risks.
8. Analyzing and classifying risks.
9. Complex project management theory.
10. Software Risk Identification.
11. Software Risk Analysis.
12. Software Risk Planning.
13. Software Risk Monitoring.
14. Software Qualitative Risk Analysis.
15. Quantitative Risk Analysis.
16. Risk management and the SDLC.
17. Risk management in CMM.

Recommended Texts

1. Dale Walter Karolak. (1995). *Software engineering risk management*. Hoboken: Wiley.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. C. Ravindranath Pandian. (2006). *Applied software risk management: A guide for software project managers*. Boca Raton: Auerbach Publications.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

SECC-7406 Advanced Requirements Engineering 3(3+0)

Requirements engineering (RE) refers to the process of defining, documenting and maintaining requirements to the sub-fields of systems engineering, software engineering concerned with this process. Despite the fundamental principles of requirements engineering being relatively well-defined, 'poor requirements' are regularly held up as the reason for project failures. Experienced business analysts know that this can be due to many factors, including failure to align requirements with business objectives, address cultural issues and document requirements at the correct level. Students will be able to understanding the enterprise, the analysis portfolio, the requirements engineering plan, The requirements taxonomy, non-functional requirements. How to write requirements, check their completeness, using models to check completeness, SRS document, Vision and Scope document, Requirement specification in agile projects, Eliciting data requirements, Modeling/specifying data requirements, Business Data Diagram, Data Dictionary, Data Flow Diagram, Software quality attributes, defining exploring and specifying software quality attributes, their relationship to functional requirements, Requirements review techniques, validating requirements against acceptance criteria.

Contents

1. Software Requirements Fundamentals: Product and process requirements, Functional and non-functional requirements, Emergent properties, Quantifiable requirements, System and software requirements.
2. Requirements Process: Process models, Process actors, Process support and management, Process quality and improvement. Requirements Analysis: Requirements sources, Elicitation techniques.
3. Requirements Analysis: Requirements classification, Conceptual modeling, Architectural design and requirements allocation, Requirements negotiation, Formal analysis.
4. Requirements Specification: System definition document, System requirements document, Software requirements specification. Requirements Validation: Requirements reviews, Prototyping, Model validation, Acceptance tests.
5. Practical Considerations: Iterative nature of the requirements process, change management, Requirements attributes, Requirements tracing, Measuring requirements. Software Requirements Tools.
6. Current research topics in requirement engineering.

Recommended Texts

1. Roger S. Pressman, Bruce R. Maxim. (2015). *Software engineering: a practitioner's approach*, (8th ed.). New York: McGraw-Hill Education.
2. Brahma Dathan, Sarnath Ramnath. (2014). *Object-oriented analysis, design and implementation, (*2nd ed.). Hyderabad: Universities Press, India.

Suggested Readings

1. Hassan Gomaa. (2011). *Software modeling and design: UML, use cases, patterns, and software architectures,* Cambridge: Cambridge University Press.
2. Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates. (2004). *Head first design patterns.* Sebastopol: O'Reilly Media, Inc.

SECC-7407 Advanced Software System Architecture 3(3+0)

Student will be able to describe the importance of predictive architecting early in the system life-cycle state/describe the applicability of model-based approaches - describe the intentionality of models throughout the system life-cycle - describe/state the characteristics and challenges of architecting system-of-systems and ultra-large-scale systems - distinguish between software architecture, enterprise architecture, system architecture, and run-time architectures. - explain the link between business strategy, business process and system- and software architecture - know of available tools for supporting architecture modelling and analysis - explain the role of architecture as a central artefact in system development. Student will be able to describe the importance of predictive architecting early in the system life-cycle state/describe the applicability of model-based approaches - describe the intentionality of models throughout the system life-cycle - describe/state the characteristics and challenges of architecting system-of-systems and ultra-large-scale systems - distinguish between software architecture, enterprise architecture, system architecture, and run-time architectures. - explain the link between business strategy, business process and system- and software architecture - know of available tools for supporting architecture modelling and analysis - explain the role of architecture as a central artefact in system development with the key objectives are to develop understanding about the fundamental architectural concepts in context of software and system and to enable students to effectively evaluate the architectures of various kinds of systems. Particular attention will be devoted to important topics, such as architectural styles and patterns, domain specific software architectures, modelling techniques, visualization and implementation of architectures, and networked and distributed architectures.

Contents

1. Quality attributes in the context of architecting.
2. Qualitative and quantitative assessment of architectures.
3. Architectural modeling through Architecture Description Languages.
4. System modeling its relation to software architecting.
5. Architecting for evolution, Partitioned and layered architectures.
6. System-of-Systems, Ultra-Large-Scale Systems.
7. Software Product Lines and Configurable Software.
8. Self-Adaptive Software, Architectural Description Languages.
9. Feature Modeling. Architecture, Model-Based Testing.
10. Current research topics in software system architecture.

Recommended Texts

1. Humberto Cervantes, Rick Kazman (2016). *Designing software architectures: A practical approach,* (1st ed.). Boston: Addison-Wesley Professional.
2. P. Clements and L. Northrup (2002). *Software product lines: practices and patterns.* Boston: Addison-Wesley.

Suggested Readings

1. R. Taylor, N. Medvidović and E.M. Dashofy (2010). *Software architecture: foundations, theory, and practice.* Hoboken: John Wiley.

SECC-7408 **Software Testing and Quality Assurance** **3(3+0)**

The only way to minimize the risk of defects while also maximizing end-user experience is by including software and quality assurance testing throughout the entire development process. Products must be tested in different ways, with different users and different scenarios to make sure that the software that end-users receive is a consistent, high-quality experience in a range of situations. The course was designed to bring focus to QA and testing since many jobs and opportunities are available in this area. The course covers both technical foundations and tools, as well as managerial and organizational aspects. More specifically, it answers questions as: What is a test case?, What is a test suite?, What is a software testing?, What is the difference between software testing and quality assurance?, What is positive and negative testing?, What is the difference between white box and black box testing?, What are the different types of testing?

Contents

1. Testing techniques.
2. Black Box testing,
3. White Box
4. Grey Box testing techniques.
5. Software system quality components and activities that support software quality
6. QA objectives: reliability, correctness, testability, maintainability, flexibility, portability, efficiency, integrity, usability, reusability, and interoperability.
7. Theoretical background: program correctness proofs, cyclomatic complexity, software reliability modeling.
8. Software unit testing to verify unit specifications
9. Integration testing to verify design specifications
10. System testing to verify requirements specifications
11. Usability testing
12. Performance, reliability, and regression testing
13. Alpha, beta and acceptance testing
14. Software testing and quality assurance tools.
15. Open source testing using NUnit and JUnit.
16. Quality Assurance planning and execution.
17. Automated testing topics include constructing a framework, scripting techniques, generating a test data, generating test architecture, pre/post-processing, test maintenance, and job specific metrics.
18. Current research topics in Software Testing and Quality Assurance.

Recommended Texts

1. Abu Sayed Mahfuz. (2016). *Software quality assurance: integrating testing, security, and audit*. Boca Raton: Auerbach Publications.
2. Mark Utting and Bruno Legeard. (2006). *Practical model-based testing: A tools approach*. San Francisco: Morgan Kaufmann Publishers Inc.

Suggested Readings

1. Jeff Tian. (2005). *Software quality engineering, testing, quality assurance, and quantifiable improvements.* Piscataway: IEEE Computer Society.
2. P Ammann and J Offutt. (2008). *Introduction to software engineering.* Cambridge: Cambridge University Press.

SECC-7409 Advanced Software Project Management 3(3+0)

To develop students' ability to plan and manage software development projects successfully, maximizing the return from each stage of the software development life cycle. This course introduces software engineers aspiring to become technical team leaders or software project managers to the responsibilities of these roles. Learn about the different types of software projects that exists and learn about the challenges that each of these types of projects poses in front of a project manager. Get to know the different and current methodologies used in the business space and know about your roles and responsibilities as a software project manager. After completion of course students will be able to learn successful software development management, including organizing the software development team; interfacing with other engineering organizations (systems engineering, quality assurance, configuration management, and test engineering); assessing development standards; selecting the best approach and tailoring the process model; estimating software cost and schedule; planning and documenting the plan; staffing the effort; managing software cost and schedule during development; risk engineering; and continuous process improvement.

Contents

1. Comparison of PM Tools.
2. PMI's Knowledge Areas
3. PMI Framework
4. PMI Process Groups.
5. Project Planning
6. Project Evaluation
7. Building Teams
8. Tracking and control
9. Comparison of Software Effort Estimation techniques
10. Comparison of different software development and designing models.
11. Analyzing different risk management methodologies.
12. Studying latest conference and journal papers related to state-of-the-art techniques.

Recommended Texts

1. Bob Hughes and Mike Cotterell (2009) Software project management, (5th ed.). New York: McGraw-Hill Education.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. Robert K. Wysocki (2011) *Effective project management: traditional, agile, extreme*, (6th ed.). Hoboken: Wiley.
2. Dwayne Phillips (2004) *The software project manager's handbook - Principles that work at work*, (2nd ed.). Piscataway: IEEE Computer Society Press and Wiley Inter-science.

CSEC-0837 **Text Processing & Data Analysis** **3 (3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing
6. Text Processing Functions.
7. Loop Control, Hashes, DBM Databases, Advanced Sorting.
8. Regular Expressions, Environment Variables, CGI-Programming.
9. Process Management, References and Data Structures.
10. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. Sebastopol: O'Reilly & Associates, Inc.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming Perl: Unmatched power for text processing and scripting*. Sebastopol: O'Reilly Media, Inc.

Suggested Readings

1. Hietaniemi, J. (2001). Cpan-comprehensive perl archive network. *Online: http://www. cpan. org*.
2. Christiansen, T., & Torkington, N. (2003). *Perl Cookbook: Solutions & Examples for Perl Programmers*. Sebastopol: O'Reilly Media, Inc.
3. Lidie, S., & Walsh, N. (2002). *Mastering Perl/Tk: Graphical User Interfaces in Perl*. Sebastopol: O'Reilly Media, Inc.

PhD

COMPUTER SCIENCE

CSEC-8401 Special Topics Natural Language Engineering 3(3+0)

This course covers a broad range of topics in computational linguistics/natural language processing, including word and sentence tokenization, text classification and sentiment analysis, spelling correction, information extraction, parsing, meaning extraction, and question answering, We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. In recent years, deep learning (or neural network) approaches have obtained very high performance across many different NLP tasks, using single end-to-end neural models that do not require traditional, task-specific feature engineering. In this course, students will gain a thorough introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Logistic Regression: Classification: The sigmoid; Learning in Logistics Regression.
2. The Cross-Entropy loss function; Gradient Descent, Regularization, etc.
3. Vector Semantics: Lexical Semantics, Vector Semantics, Words and Vectors.
4. Cosine for measuring similarity, TF-IDF Weighting terms in the vector, etc.
5. Neural Networks and Neural Language Models.
6. Units, The XOR problem, Feed Forward Neural Network, Training Neural Nets, etc.
7. Part-of-Speech Tagging.
8. Word Classes, Penn POS Tagset, POS Tagging, HMM POS Tagging, etc.
9. Sequence Processing with Recurrent Networks Simple recurrent neural network, Applications of Recurrent Neural Networks.
10. Deep Networks: Stacked and Bidirectional RNNs, Managing Context in RNNs: LSTMs and GRUs, Words, Subwords and Characters.
11. Encoder-Decoder Models.
12. Attention, and Contextual Embeddings

Recommended Texts

1. Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. New York: MIT press.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

CSEC-8402 Special Topics in Grammar Engineering 3(3+0)

Natural language processing (NLP) enables computers to make use of data represented in human language (including the vast quantities of data available on the web) and to interact with computers on human terms. Applications from machine translation to speech recognition and web-based information retrieval demand both precision and robustness from NLP technology. Meetings these demands will require better hand-built grammars of human languages combined with sophisticated statistical processing methods. This course focuses on the implementation of linguistic grammars, drawing on a combination of sound grammatical theory and engineering skills. This course introduces a basic knowledge of key syntactic concepts, such as word classes, constituency and phrase structure and introduces the key components of a major theory of syntax: Lexical Functional Grammar by way of intro to LFG but plenty on structural analysis that will be helpful. Class meetings will alternate between lectures and hands-on lab sessions. We will cover the implementation of constraints in morphology, syntax and semantics within a unification-based lexicalist framework of grammar.

Contents

1. Introduction, LFG, Templates, C & F description, Agreement, Determiners, Rules & alternations, Adjuncts, Obliques, Prepositions, Pronouns, Punctuation, Generation & Optimality, Complements, Uncertainty, Imperatives, Finite-State Morphology, Free Word Order and the Shuffle Operator, Coordination
2. Introduction and Overview, LFG Basics.
3. LFG Basics II, Templates I, MacOSX, Unix.
4. Templates II, f-descriptions, Subject-Verb Agreement, Determiners, xlerc file
5. Lexical Rules, Passive and Argument alternations.
6. Adjuncts (Adjectives and Adverbs) and Obliques: PPs, Semantic and Non-Semantic Prepositions.
7. Pronouns, Lexical Entries, Punctuation, Note on Adjuncts: Sets and Scope.
8. Generation & Optimality Projection, Restricting Over-generation
9. Complements, xcomp and comp.
10. Functional Uncertainty, Imperatives and empty categories.

Recommended Texts

1. Butt, M., King, T. H., Nino, M. E., & Segond, F. (1999). *A grammar writer's cookbook*. Stanford: CSLI. Publications.
2. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.

Suggested Readings

1. Dalrymple, M. (2001). *Lexical functional grammar*. Netherlands: Brill.
2. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.

CSEC-8403 Advanced Topics in Formal Methods 3(3+0)

The objective of the course is the study of the most important among the qualities of software: correctness. Correctness will be studies looking at the conceptual perspective as well as the realization perspective. Modelling and verification of both static (data) and dynamic (processes) aspects will be considered. The various topics will be treated emphasizing methodological, theoretical and practical facets. The course will introduce various forms of logic (first-order logic, temporal logics, fix point logics), techniques and tools for automated verification. After the successful completion of the course the student will have acquired techniques and methods for proving correctness of programs and conceptual models. The course presents advanced techniques for formal specification of requirements, designs and implementations of software-based systems, focusing on software architecture as the organising principle for software development. Course content will include some topics amongst: comparative properties of formalisms for specification; meta-properties of specification formalisms, including interpolation properties, modularity and their relationships; component-based approaches and the role of category theory in component composition; formalisation of encapsulation, cohesion and coupling; externalisation of interaction definition; formal toolkit for software architecture.

Contents

1. First Order Logic and Set theory. Set theory and first order logic form the foundations of mathematics and formal reasoning. Syntax and semantics of first-order logic, basic model theory, and basic proof theory.
2. Godel's completeness theorem, the compactness theorem, the Lowenheim-Skolem theorems, Godel's incompleteness theorem, and non-standard models.
3. Aspects of axiomatic set theory. Logic used to model and reason about computation.
4. Un-decidability and Decision Procedures. Automate proof discovery and mathematical reasoning. We will cover basic un-decidability results.
5. Decision procedures, including the DPLL algorithm for deciding SAT (Boolean Satisfiability) and the SMT (Satisfiability modulo theories) framework.
6. Reactive Systems. Various temporal logics and calculi, automata on infinite objects, the notions of safety and liveness, and the theory of refinement.
7. Tools and Applications. Current verification tools
8. Interactive theorem provers, SMT solvers, and model checkers.
9. Applications to hardware verification, software verification, databases, and security.

Recommended Texts

1. Robinson, A. J., & Voronkov, A. (Eds.). (2001). *Handbook of automated reasoning*. Houston: Gulf Professional Publishing.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Suggested Readings

1. Cousot, R. (2005). *Verification, model checking, and abstract interpretation*. Berlin: Springer
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

CSEC-8404 Special Topics in Cluster & Grid Computing 3(3+0)

The purpose of the course is to provide basic knowledge on the most important principles, methods, tools, systems, standards, etc. behind these two evolving basic technologies. This course will introduce you to the multiple forms of parallelism found in modern Intel architecture processors and teach you the programming frameworks for handling this parallelism in applications. This course can apply to various HPC and datacenter workloads and framework including artificial intelligence (AI). You will learn how to handle data parallelism with vector instructions, task parallelism in shared memory with threads, parallelism in distributed memory with message passing, and memory architecture parallelism with optimized data containers. This knowledge will help you to accelerate computational applications by orders of magnitude, all the while keeping your code portable and future-proof. Prerequisite: programming in C/C++ or Fortran in the Linux environment and Linux shell proficiency (navigation, file copying, editing files in the text-based editors, compilation).

Contents

1. Introduction to distributed and high-performance computing. Basic terms: distributed computing, HPC, HPCC, network computing, Internet computing, cluster, grid, meta-computing, middleware, etc; milestones of the history, some representative applications.
2. Classification: Taxonomies, MPP, SMP, CC-NUMA,
3. Cluster: dedicated high performance (HP), high availability (HA), CoPs, PoPs, CoWs; distributed, on-demand, high-throughput, collaborative, data-intensive computing.
4. Basics of communication media and protocols: TCP/IP, Internet2, QoS, ATM, Fast Ethernet, etc.
5. Programming models: Message passing, client-server, peer-to-peer, broker computing, code shipping, proxy computing, mobile agents. Toolkit and OO systems.
6. Higher level communication: Light-weight communication, sockets, standard APIs, active messages. Storage and file problems: Network RAM, RAID and software RAID.
7. Distributed File systems: NFS, AFS, OSF-DSF, RSF. Message passing standards: PVM (Parallel Virtual Machine), MPI (Message Passing Interface). Object-oriented defacto standards CORBA and DCOM. Java-based methods: JVM, RMI, Bytecode, Applet and Servlet, JavaBean and JavaSpaces, Jini.
8. Grid toolkit approach: Globus Hourglass concept, communication, resource and process management, data access, security. Object-oriented approach: Legion Language support, component wrapping, program support, resource management.
9. Security: Confidentiality, integrity and availability. Authentication, authorization, assurance, auditing, accounting.
10. Scheduling: Algorithms, policies and techniques, high performance and high throughput schedulers, resource scheduling. Grid monitoring: Tasks, types, architecture, components.

Recommended Texts

1. Wilkinson, B. (2009). *Grid computing: techniques and applications*. Boca Raton: CRC Press.
2. Buyya, R. (1999). *High performance cluster computing: architectures and systems*. Upper Saddle River, NJ: Prentice Hall,

Suggested Readings

1. Kacsuk, P., Lovas, R., & Namath, Z. (Eds.). (2008). *Distributed and parallel systems: in focus: desktop grid computing*. Berlin: Springer Science & Business Media.
2. Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

CSEC-8405 **Advanced Topics in Parallel Processing** **3(3+0)**

The major objective of this course is to develop structural intuition of how the hardware and the software work, starting from simple systems to complex shared resource architectures; to provide guidelines about how to write and document a software package; to familiarize the audience with the main parallel programming techniques and the common software packages/libraries. This is a graduate level course on parallel computing with the objective to familiarize students with the fundamental concepts, techniques and tools of parallel computing. Participation in this course will enable you to better use parallel computing in your application area, and will prepare you to take advanced courses in more specific areas of parallel computing. Upon completion of this course you will; be able to understand and employ the fundamental concepts and mechanisms which form the basis of the design of parallel computation models and algorithms; be able to recognize problems and limitations to parallel systems, as well as possible solutions; and be familiar with some of the relevant papers in the area of parallel algorithms and systems, both current and classic.

Contents

1. Parallel and systolic computations
2. Architectures of parallel processors
3. VLSI computational networks
4. Languages and programming environments for parallel systems
5. Design and analysis of parallel
6. Systolic algorithms
7. Reconfigurable and data driven processor arrays
8. Complexity measures of VLSI computations
9. Application of parallel processors to supercomputing.

Recommended Texts

1. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.
2. Wilkinson, B. (2009). *Grid computing: techniques and applications*. Boca Raton: CRC Press.

Suggested Readings

1. Roosta, S. H. (2012). *Parallel processing and parallel algorithms: theory and computation*. Berlin: Springer Science & Business Media.
2. Kacsuk, P., Lovas, R., & Namath, Z. (Eds.). (2008). *Distributed and parallel systems: in focus: desktop grid computing*. Berlin: Springer Science & Business Media.

CSEC-8406 **Special Topics in Distributed Computing** **3(3+0)**

Be able to understand and develop distributed computing systems. Topics include principles of naming and location, atomicity, resource sharing, concurrency control and other synchronization, deadlock detection and avoidance, security, distributed data access and control, integration of operating systems and computer networks, distributed systems design, consistency control, and fault tolerance. Distributed systems help programmers aggregate the resources of many networked computers to construct highly available and scalable services. This class teaches the abstractions, design and implementation techniques that enable the building of fast, scalable, fault-tolerant distributed systems. Topics include multithreading, network programming, consistency, fault tolerance, consensus, security, and several case studies of distributed systems. The main objective of this course is to provide learners with a solid foundation for understanding, and specifying distributed services, and designing and analyzing distributed algorithms for reliable and fault-tolerant implementations of these distributed services. The services studied are typical in modern data-centers and cloud computing infrastructures.

Contents

1. Building correctly functioning, performance-oriented, reliable and secure distributed systems, especially in web-based environments using Java and XML;
2. Study the design principles of distributed systems and their application to the modern networked environment;
3. Fundamental distributed systems theory including group communication, synchronization, concurrency control, load balancing and scheduling, replication, fault-tolerance, and network security;
4. Client/server systems, middleware and middleware-based systems, network computing, and networked file systems.
5. Replication, fault tolerance, consistency, scalability, isolation and privacy in large-scale distributed systems.
6. In-depth coverage of cloud computing strategies to process voluminous data.

Recommended Texts

1. Coulouris, G. F., Dollimore, J., & Kindberg, T. (2005). *Distributed systems: concepts and design*. London: Pearson Education.
2. Mullender, S. (1993). *Distributed systems*. New York: ACM press.

Suggested Readings

1. Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Upper Saddle River, NJ: Prentice-Hall.

2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8407 **Special Topics in Open Source Software** **3(3+0)**

Throughout the last decade, the phenomenon of open source software has evolved from what was once a technical curiosity into a mainstay of routine business practice. The emergence of production quality systems such as Linux, MySQL, and Apache have allowed for open source technologies to be viable options for firms choosing to develop their own information technology infrastructure. The distinctive nature of open source communities, firms, and technologies, however, gives rise to issues that are profoundly different from those encountered when a system is acquired from a traditional vendor. In this course, we will define what open source software is, we will go over the history of open source software and what benefits it has provided to the world's technology infrastructure over the decades. The course aims to teach you how to work in open source projects productively and gain a lot of benefits from it, and really understand what the advantages are. You will learn about collaboration best practices, and how to encourage diversity in open source projects. We will discuss the different licensing models that are available, as this will help you decide what the best license is for your project.

Contents

1. Issues associated with open source technologies, with a focus on understanding the implications for businesses that are interested in using them.
2. Through a combination of readings, presentations, discussion and hands-on projects.
3. We will examine: the characteristics of key open source technologies (Linux, MySQL, Apache, et al).
4. The nature of open source communities, their development processes.
5. Evolving structure of the open source industry.

Recommended Texts

1. Fink, M. (2003). *The business and economics of linux and open source*. New York: Prentice Hall Professional.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

Suggested Readings

1. Raymond, E. (2002). *The cathedral and the bazaar: musings on linux and open source by an accidental revolutionary*. California: Sebastopol.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

.

CSEC-8408 **Advanced Research Methods** **3(3+0)**

The primary objective of this course is to develop a research orientation among the scholars and to acquaint them with fundamentals of research methods. Specifically, the course aims at introducing them to the basic concepts used in research and to scientific social research methods and their approach. It includes discussions on sampling techniques, research designs and techniques of analysis. The course is designed to aid graduate students to improve their writing and research skills and is specific to criminal justice. It includes an overview of the writing mechanics and process, how to properly conduct research, cite sources and guidelines for writing specific types of criminal justice papers. This course equips students with the necessary knowledge and skills to complete a large-scale dissertation project required for a Master's award in research. This course improves students' abilities in several key areas, enabling them to develop an advanced understanding of various research philosophies, methods, data analyses and presentation styles.

Contents

1. What is Research? Types of Research, Why Research, Significance & Status of Research in Computer Science. Steps in Research: Having grounding in Computer Science, Major Journals & Publication in Computer Science, Major Research areas of Computer Science.
2. Hypothesis formulation, developing a research proposal, how engineering research differs from scientific research, The role of empirical studies.
3. Domain Theories, and Modeling Assumptions, Qualitative Reasoning Techniques, Model Formulation, Causal Reasoning, Simulation, Comparative Analysis, Teleological Reasoning, Data Interpretation, Planning, Spatial Reasoning.
4. Simulation: What is simulation? How a simulation model works? Time & randomness in simulation.
5. Applications of simulations. Research Data: What is data, Mathematical statistics and computer science views on data analysis, Methods for finding associations: regression and pattern recognition.
6. Method for aggregation and visualization: principal components and clustering, Hypothesis testing. Literature Survey: Finding out about your research area, Literature search strategy, writing critical reviews, identifying venues for publishing your research.
7. Writing Papers and the Review Process: Preparing and presenting your paper. The conference review process, making use of the referees' reports, the journal review process, Group exercise in reviewing research papers.
8. Ethical issues and Professional Conduct Ethics in general, Professional Ethics, Ethical Issues that Arise from Computer Technology, General Moral Imperatives, More Specific Professional Responsibilities, Organizational Leadership Imperatives.

Recommended Texts

1. Dane, F. C. (1990). *Research methods*. CA: Brooks/Cole Publishing.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

Suggested Readings

1. Ahuja, R. (2001). *Research methods*. Jaipur: Rawat publications.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8409 **Evolutionary Computation** **3(3+0)**

Evolutionary Computation can be considered as a sub-field of Artificial Intelligence. Evolutionary algorithms use Nature as a metaphor and are inspired in the principles of natural selection and genetics. These algorithms have been applied successfully for solving difficult problems across a broad spectrum of fields, including engineering, economics and finance, architecture, design, automatic programming, art generation, and many others. In this course, you will learn the basic working principles of these algorithms. Advance topics on Evolutionary Computation are also under discussion. Also, scholar will able to learn how to Parallelize GA, ACO and Genetic Algorithms? Limitations of simple EAs how they can be scalable, Bayesian belief networks and fuzzy belief networks. Moreover, this course will cover major methods for selection, recombination, mutation, and replacement. The representations and design of operators with local and global searching methods. Also, different optimization algorithms such as PSO, Grey Wolf, Blue Whale optimization, Honey bee, cat swarm optimization are included in this course.

Contents

1. What is Evolutionary Computation? Historical perspective, Limitations of simple EAs
2. Major classes of Evolutionary Algorithms.
3. Local vs global search methods.
4. Different types of genetic algorithms.
5. Major methods for selection, recombination, mutation, and replacement.
6. Representations and design of operators.
7. Evolution strategies.
8. Genetic programming and Genetic Algorithms
9. Swarm Intelligence and Its variations
10. Ant Colony Optimization and Its variations
11. Support Vector Machines
12. Probabilistic reasoning: Bayesian reasoning and Dempster-Shafer theory
13. How to Parallelize GA, ACO and Genetic Algorithms?

Recommended Texts

1. David E. (2005). *Genetic algorithms in search, optimization and machine learning.* Boston: Addison-Wesley
2. Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Berlin: Springer.

Suggested Readings

1. Karray, F., Karray, F. O., & De Silva, C. W. (2004). *Soft computing and intelligent systems design: theory, tools, and applications*. London: Pearson Education.
2. Engelbrecht, A. P. (2007). *Computational intelligence: an introduction*. Hoboken: John Wiley & Sons.

CSEC-8410 Special Topics in Machine Learning 3(3+0)

This course covers several modern machine-learning topics at an advanced level. The current list of possible topics includes deep learning, maximum-margin methods, kernel methods, graphical models and Monte Carlo methods. The theoretical and mathematical underpinnings of these topics will be discussed along with their role in modern machine learning by reading and discussing recent publications. In this course, student will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work. More importantly, student will learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. The scholar will able to learn probabilistic graphical models: bayesian belief networks probabilistic graphical models: learning from data probabilistic graphical models: em algorithm collaborative filtering, maximum-margin methods hidden markov models and deep learning. Multi class classification, different models, and features selection are also included in this course.

Contents

1. Introduction to machine learning.
2. Generative and Discriminative Learning
3. Graphical Models: Representations, Learning and Inference, Linear regression
4. Gradient Descent as general parameter Learning/optimization approach.
5. Logistic regression, One-vs-all classification, Regularization.
6. Neural Networks.
7. Practical advice for applying learning algorithms: How to develop, debugging, feature/model design, setting up experiment structure
8. Probability and Bayesian Analytics.
9. Classification Algorithm
10. Naive Bayes.
11. Support Vector Machines (SVMs) and the intuition behind them.
12. Unsupervised learning: clustering and dimensionality reduction.
13. Anomaly detection.
14. Probabilistic Graphical Models: Bayesian Belief Networks
15. Probabilistic Graphical Models: Learning from Data
16. Probabilistic Graphical Models: EM algorithm
17. Hidden Markov Models
18. Deep Learning

Recommended Texts

1. Tom Mitchell, (2012), *Machine learning*. New York: McGraw-Hill.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Suggested Readings

1. Trevor Hastie and Robert Tibshirani, (2016) *The elements of statistical learning*. Berlin: Springer.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8411 **Special Topics in Artificial Intelligence** **3(3+0)**

The students should obtain advanced theoretical knowledge and technical skills on the topics covered in the course. The course will enable students to carry out advanced research projects on topics within artificial intelligence and apply the knowledge to complex intelligent system design and development. Upon completion of the course the candidate should be able to discuss advanced theoretical and technical issues in artificial intelligence. Apply knowledge for complex intelligent system design and development. Carry out advanced research on artificial intelligence. Also learn evolutionary computation: simulated annealing, genetic algorithms, genetic, genetic programming, natural language processing: history of nlp, syntax and formal grammars, syntax and formal grammar, statistical parsing, hidden markov model, wordnet, question answering system, agent and multi-agent systems automated planning: planning terminology. Planning as search, hierarchical planning, case-based planning. Planning approaches to learning system. Selective topic on cognitive science, selective topic on neuro-computing. Also, different learning models are part of this course.

Contents

1. Introduction: Turing Test, Strong AI vs Weak AI, Heuristics, Applications and Methods.
2. Uninformed Search: Search in IS, Genrate-and-Test Paradigm, Blind Search Algorithm.
3. Informed Search: Heuristics, the Best-First Search, The Beam Search, The A* Search.
4. Search Using Game: Game Trees and Minimum, Game Theory.
5. Logic in AI: Logic and Representation, Propositional Logic, Predicate Logic, Other Logics.
6. Knowledge Representation: Search Tree, Production System, Objects, Frames, Scripts & the Conceptual Dependency System, Semantic Networks, Recent Approaches, Agents.
7. Production Systems: Strong Methods vs. Weak Methods
8. Uncertainty in AI: Fuzzy Sets, Fuzzy Logic, Fuzzy Inference
9. Probability Theory and Uncertainty.
10. Expert Systems: Characteristics of ES, Knowledge Engineering.
11. Knowledge Acquisition, Classical ES, Case-Based Reasoning.
12. Neural Networks: Introduction, The Perceptron Learning Rule, Back propagation, Discrete Hopfield Networks, Application Areas.

Recommended Texts

1. Russell, S., & Norvig, P. (2002). *Artificial intelligence: a modern approach*. Upper Saddle River, New Jersey: Prentice Hall
2. Lucci, S., & Kopec, D. (2015). *Artificial intelligence in the 21st century*. Sterling: Stylus Publishing.

Suggested Readings

1. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8412 Advanced Optimization Methods 3(3+0)

This course introduces the principal algorithms for linear, network, discrete, nonlinear, dynamic optimization and optimal control. Emphasis is on methodology and the underlying mathematical structures. Topics include the simplex method, network flow methods, branch and bound and cutting plane methods for discrete optimization, optimality conditions for nonlinear optimization, interior point methods for convex optimization, newton's method, heuristic methods, and dynamic programming and optimal control methods. This course will also cover special topics on discrete optimization, special topics on dynamic optimization and special topics on nonlinear optimization. Also, equality and inequality constraints, sumt method of fiacccco and mccormick, linear optimization, robust optimization and network flows topics are included. Moreover, search methods for functions of *n* variables: method of hooke and jeeves, nelder and mead’s method. Gradient methods: davidon-fletcher-powell (dfp) and quadratic function and hessian matrix; uniqueness of minimum are part of this course. A term paper is also part of this course.

Contents

1. Preliminaries: Review of the theory of maxima, minima (two variables); positive definite matrices, convexity of regions and functions;
2. Quadratic function and Hessian matrix; uniqueness of minimum.
3. Classical methods for functions of one variable and *n* variables, Newton’s method
4. Search methods for functions of one variable: Single search techniques:
5. Bracketing method; Quadratic and cubic interpolation;
6. Fibonacci search; Golden-section.
7. Search methods for functions of *n* variables: method of Hooke and Jeeves,
8. Nelder and Mead’s Method.
9. Gradient methods: Davidon-Fletcher-Powell (DFP);
10. Fletcher- Reeves, conjugate- gradient and direct- search methods,
11. Newton’s method, method of Steepest descent.
12. Review of Lagrange multipliers technique with equality constraints;
13. Inequality constraints and slack variables; Kuhn-Tucker conditions.
14. Search methods: modified Hooke and Jeeves,
15. The Complex method.
16. Penalty-function approach to constrained optimization;

Recommended Texts

1. Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization*. Belmont: Athena Scientific.
2. Bonnans, J. F., Gilbert, J. C., Lemaréchal, C., & Sagastizábal, C. A. (2006). *Numerical optimization: theoretical and practical aspects*. Berlin: Springer Science & Business Media.

Suggested Readings

3. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.
4. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8413 **Special Topics in Data Mining** **3(3+0)**

Data Mining is one of the hottest fields in Computer Science. Data has been accumulating throughout the computer age in many forms, including database systems, spreadsheets, text files, and recently web pages. Data mining aims to search through data for hidden relationships and patterns in your data. This is a special topic course on data mining. We will cover advanced topics such as large-scale data mining using Map-reduce, similarity search (including minwise hashing and locality sensitive hashing), mining data streams, mining social networks, relational data mining, and matrix factorization methods for data mining. This course will be highly beneficial to students whose research interests are in database, data mining, bio-informatics, information retrieval, decision science and artificial intelligence, and also to those who may need to apply data mining in any diversified application area. At the end of this course, the scholar will become familiar with the fundamental concepts of data mining and analytics; will become competent in recognizing challenges faced by applications dealing with very large volumes of data as well as in proposing scalable solutions for them; and will be able to understand how business intelligence impact scientific discovery, and our day-to-day life.

Contents

1. Basic Algorithms discussion in few lectures than moving on to advance and special topics
2. Large-scale Data Mining
3. Mining Data Streams
4. Mining Social Networks
5. Relational Data Mining
6. Tree/Graph Mining
7. Privacy-preserving Data Mining
8. High-Dimensional Data Clustering
9. Large Scale Classification Algorithms
10. Web Applications (including advertising, recommendation, and summarization).

Recommended Texts

1. Han, J., Kamber, M., & Pei, J. (2011). *Data mining concepts and techniques* (3rd ed.). Massachusetts: Morgan Kaufmann.
2. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive data sets*. Cambridge: Cambridge university press.

Suggested Readings

1. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.
2. Padua, D. (2011). *Encyclopedia of parallel computing*. Berlin: Springer Science & Business Media.

CSEC-8414 Peer-To-Peer Systems 3(3+0)

Peer-to-Peer Networking (APPN), part of IBM's Systems Network Architecture (SNA), is a group of protocols for setting up or configuring program-to-program communication within an IBM SNA network. Using APPN, a group of computers can be automatically configured by one of the computers acting as a network controller so that peer programs in various computers will be able to communicate with other using specified network routing. Peer-to-peer systems have recently gained a lot of attention in the social, academic, and commercial communities. One of the early driving forces behind the peer-to-peer concept is that there are many PCs in homes and offices that lie idle for large chunks of time. By the end of course students will be able to understand the principles and desired properties of distributed systems based on different application areas, understand and apply the basic theoretical concepts and algorithms of distributed systems in problem solving, recognize the inherent difficulties that arise due to distributed-ness of computing resources, identify the challenges in developing distributed applications.

Contents

1. Overview of P2P Systems and brief history
2. Taxonomy of P2P Networks/Systems and Analysis of popular P2P Systems
3. Analysis of unstructured P2P Systems
4. Analysis of structured P2P Systems
5. P2P Network Architecture
6. Topology Control
7. Incentive issues in P2P Systems
8. Search Efficiency
9. P2P-based content delivery
10. Security and Reliability
11. Replication in peer-to-peer systems
12. Anonymity in peer-to-peer systems
13. Distributed network control
14. Dynamic peer-to-peer exchange, Peer-to-Peer network configuration, Decentralized peer-to-peer network, Social, Legal and Privacy aspects of P2P Systems.

Recommended Texts

1. Yu-Kwong, R. k. (2012) *Peer-to-peer computing: applications, architecture, protocols, and challenges*. Ohio: CRC Press.
2. Ralf, S. & Klaus, W. (2005). *Peer-to-peer systems and applications*. Berlin: Springer.

Suggested Readings

1. Xuemin, S. Heather, Y. John, B. & Mursalin, Akon. (2010) *Handbook of peer-to-peer networking*. Berlin: Springer.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8415 **Ubiquitous Information Interaction** **3(3+0)**

The purpose of this course is to introduce students with: the objectives and the historical development of the field of ubiquitous computing; development in new materials; fundamentals of sensor technology and sensor networks; design of new (often embedded) interactive artefacts; context aware and adaptive systems; middleware for fine-grained distributed systems; analysis and coordination of complex systems; new styles of interaction, e.g. Tangible interfaces; most important applications in the field; general implications of the field. Intelligent system interaction, autonomous systems and artificial life and ubiquitous communication. At the end of this course student will able to understand different concept of mobile technology in the messy now; infrastructure; seams, seamlessness, seamfulness, evaluating interaction of ubicomp systems. Illustrative ubiquitous computing applications, modelling the key ubiquitous computing properties, smart devices and services, smart mobiles, cards and device networks, information interaction, seminal ideas of ubiquitous computing tangibility and embodiment. Also, this course includes a term paper.

Contents

1 Information Interaction
2 Seminal ideas of ubiquitous computing
3 Tangibility and Embodiment
4 Social computing; Privacy
5 Critical and cultural perspectives
6 Mobility and Spatiality
7 Mobile Technology in the Messy Now; Infrastructure; Seams, seamlessness, seamfulness
8 Evaluating Interaction of Ubicomp systems.
9 Illustrative Ubiquitous Computing Applications
10 Modelling the Key Ubiquitous Computing Properties
11 Smart Devices and Services
12 Smart Mobiles, Cards and Device Networks

Recommended Texts

1. Stefan, P. (2011). *Ubiquitous computing: smart devices, environments and interactions*. Hoboken: John Wiley & Sons.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1 Rafael, A. B. (2008). *Bringing iterative design to ubiquitous computing: interaction techniques*. Gottingen: Cuviller Verlag.
2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8416 Evolution and Re-Engineering 3(3+0)

The objective of the course is to examine the set of tools and techniques that allow a smooth re-engineering of old legacy systems to new more maintainable systems that use more appropriate and robust advanced technologies (i.e. Object-oriented, network-centric). The scholars will be able to learn software infrastructure, architecture transformation, code and data migration, integration plan and architecture recovery. This course will also include the topics related to evolution activities, reverse engineering, re-factoring, program transformation, visualization; re-engineering techniques: code restructuring and source code analysis architecture recovery risk managed modernization software infrastructure maintaining transactional context architecture transformation code and data migration integrated plan topics in re-engineering research. After completion of this course the scholars will able to learn introduction: challenges of evolution legacy systems, laws of software evolution, evolution models, testing in the context of evolution, metrics for evolution, evolution activities: reverse engineering, re-factoring, program transformation, visualization; re-engineering techniques: code restructuring and source code analysis.

Contents

1. Introduction: Challenges of evolution
2. Legacy systems
3. Evolution Process: Laws of software evolution
4. Evolution models
5. Testing in the context of evolution
6. Metrics for evolution
7. Evolution Activities: Reverse engineering, Re-factoring, Program Transformation, Visualization;
8. Re-engineering techniques: Code restructuring and Source code analysis
9. Risk Managed Modernization
10. Software Infrastructure
11. Maintaining Transactional Context
12. Architecture Transformation
13. Code and Data Migration
14. Integrated Plan, Topics in re-engineering research.

Recommended Texts

1 Daniel, P. Grace, A. L. & Robert C. S. (2003). *Modernizing legacy systems: software technologies, engineering processes, and business practices.* Boston: Addison-Wesley.
2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1 Martin, F. & Kent, B. (2000). *Refactoring: improving the design from existing code*. Boston: Addison-Wesley.
2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8417 Program Comprehension and Reverse Engineering 3(3+0)

The objective of this course is to introduce students with reverse engineering techniques and algorithms. Reverse engineering brings together in one place important contributions and up-to-date research results in this important area. Reverse engineering serves as an excellent reference, providing insight into some of the most important issues in the field. These topics are included reverse engineering: design recovery and re-documentation challenges in reverse engineering, reverse engineering approaches reverse engineering techniques, reverse engineering at the program level, reverse engineering at the architectural level, reverse engineering tools, piracy and copy protection anti-reversing techniques breaking protections. After completion of course students also will be able to learn static analysis: parsing, lexical analysis, issues in parsing languages program analysis: control flow analysis, data flow analysis, flow graphs, program dependence graphs, call graphs and dynamic analysis: profiling, dynamic testing and design recovery and re-documentation and reverse engineering techniques. Also, some aspects of piracy and copyright protection.

Contents

1. Understanding Interleaved Code
2. Pattern Matching for Clone and Concept Detection
3. Extracting Architectural Features from Source Code
4. Static Analysis: Parsing, lexical analysis, issues in parsing languages
5. Program analysis: Control flow analysis, Data flow analysis, flow graphs, program dependence graphs, call graphs
6. Dynamic analysis: Profiling, dynamic testing
7. Reverse engineering: Design recovery and re-documentation
8. Challenges in reverse engineering
9. Reverse engineering approaches
10. Reverse engineering at the program level
11. Reverse engineering at the architectural level
12. Reverse Engineering tools.
13. Piracy and Copy Protection
14. Antireversing Techniques, Breaking Protections

Recommended Texts

1. Wills, L. M. & Philip N. (2007) Reverse engineering. New York: Springer Science & Business Media.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Eldad, E. (2011). *Reversing: Secrets of reverse engineering*. Hoboken: Wiley Publishing.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8418 **Software Refactoring** **3 (3+0)**

The objective of this course is to learn the techniques for: use refactoring to facilitate adding new functionality to system; use refactoring to improve design; refactor existing applications to make them more maintainable; use tests during refactoring; recognize when and when not to refactor; identify and choose the appropriate type of refactoring technique to solve specific problems. The scholar will able to learn, refactoring principles, reasons for refactoring, what to refactor, challenges in refactoring categories, refactoring in the small and large, refactoring techniques, recognizing bad smells in code refactoring for organizing code, higher abstraction, improvement and others, refactoring of UML models, tool support for refactoring: strengths and limitations, big refactoring, refactoring, reuse, and reality. After completion of course students will be able to learn successful software refactoring and evolution, Big Refactorings, Refactoring, Reuse, and Reality in software systems. How a systematically a software can be modified and strengths and limitations of refactoring.

Contents

1. Refactoring principles
2. Reasons for refactoring
3. What to refactor
4. Challenges in refactoring
5. Refactoring categories
6. Refactoring in the small and large
7. Refactoring techniques
8. Recognizing bad smells in code
9. Refactoring for organizing code
10. Higher abstraction
11. Improvement and others
12. Refactoring of UML models
13. Tool support for refactoring: Strengths and limitations
14. Big Refactorings, Refactoring, Reuse, and Reality

Recommended Texts

1. Martin, F. Kent, B. John, B. William, O. Don, R. (2018) *Refactoring: improving the design of existing code.* Boston: Addison-Wesley.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1 Martin, L. Stephen, R. (2006). *Refactoring in large software projects: performing complex restructurings.* Hoboken: Wiley Publishing.

2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8419 **Advance Software Architecture** **3 (3+0)**

This course enables the students to understand the challenges of advanced software design and the issues associated with large-scale software architectures, frameworks, patterns and components. To develop the students' understanding of the tools and techniques that may be used for the automatic analysis and evaluation of software. Scholars are able to learn and understand architectural analysis, Comparison of methods, Modeling and analyzing dynamic software architectures, Service oriented architectures, software product lines, testing line, and their quality attributes. Tools for architecture reconstruction, evolution and reconstruction are also discussed. Different methods are compared based on architecture, evaluation criteria, metrics evaluation, self-healing architecture and on the quality attributes. Latest research articles are also part of course related to architecture construction, architectural evaluation, Service oriented architectures, aarchitectures in dynamic environments, self-healing approaches, component re-use in architecture line. Different models for software construction, evolution, analysis and studied and compared. Term paper is also part of this course.

Contents

1 Re-use in architectures
2 Software product lines
3 Evaluation and validation of product lines
4 Product line testing
5 Re-use in product lines
6 Service oriented architectures (SOAs)
7 SOA concepts, risks and challenges
8 Quality attributes and SOAs
9 Evaluating and testing SOAs
10 Architectural evaluation
11 Methods for architectural analysis, Comparison of methods
12 Architectural evolution and reconstruction

Recommended Texts

1 Taylor, R. Medvidović, N. & Dashofy, E.M. (2010) Software architecture: foundations, theory, and practice. Hoboken: John Wiley.

2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1 Clements and L. Northrup. (2002). Software product lines: practices and patterns. Boston :Addison-Wesley.

2 Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8420 **Category Theory** **3 (3+0)**

Category theory is an emerging area and a challenging subject in the computer science. This course aims to develop students' basic understanding in category theory so that they may be able to develop scientific models using category theory, especially in the field of computer science. This course introduces basic concepts required for learning this emerging field in the computer science. The course aims at familiarizing the students with five major ideas of category theory: categories, functors, natural transformations, universality, and adjoints. Moreover, this course includes mappings, natural transformations and the algebras, introduction to category theory, category of sets, sets and functions, commutative diagrams, ologs, products and coproducts, finite limits in set, finite colimits in set, and other notions in set. Also, this course includes study of monoids, groups graphs, orders databases: schemas and instances, basic category theory, categories and factors commonly arising in mathematics, natural transformations, categories and schemas equivalence.

Contents

1. Basic concepts of category theory
2. Mappings
3. Natural transformations and the algebras.
4. Introduction to category theory
5. Category of sets
6. Sets and functions
7. Commutative diagrams
8. Ologs
9. Products and coproducts
10. Finite limits in Set
11. Finite colimits in Set
12. Other notions in Set
13. Categories and functors
14. Monoids, Groups
15. Graphs, Orders
16. Databases: schemas and instances
17. Basic category theory
18. Categories and functors commonly arising in mathematics
19. Natural transformations, Categories and schemas equivalence
20. Limits and colimits, Other notions in Cat
21. Categories at work: Adjoint functors, Categories of functors, Monads, Operads

Recommended Texts

1. Spivak, D. I. (2013). *Category theory for scientists*. Cambridge: The MIT Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Awodey, S. (2010). *Category theory*. Oxford: Oxford University Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8421 **Concrete Mathematics** **3 (3+0)**

Concrete mathematics is useful for advanced computer programming and the analysis of algorithms. This course aims to develop a solid and relevant base of mathematical skills to solve complex problems, and to discover subtle patterns in data. Concrete mathematics deals with continuous and discrete mathematics. Major topics include: sums, recurrences, integer functions, elementary number theory, binomial coefficients, generating functions, discrete probability, asymptotic methods. This course includes mechanical summation special numbers: stirling numbers, eulerian numbers, harmonic numbers, harmonic summation, bernoulli numbers, fibonacci numbers continuants, generating functions: domino theory and change, basic maneuvers, solving recurrences, special generating functions, convolutions, exponential generating functions, dirichlet generating functions. Also, this course covers the basics of integer functions: floors and ceilings, floor/ceiling applications, floor/ceiling recurrences, 'mod': the binary operation, floor/ceiling sums. Some concepts of hypergeometric functions, hypergeometric transformations, partial hypergeometric sums are also part of this course. In this course a discussion is included about general methods of problem solving.

Contents

1. Recurrent problems, sums, integer functions
2. Binomial coefficients
3. Special numbers and generating functions.
4. Recurrent Problems: The Tower of Hanoi
5. Lines in the Plane
6. The Josephus Problem
7. Sums: Notation, Sums and Recurrences
8. Manipulation of Sums, Multiple Sums
9. General Methods
10. Finite and Infinite Calculus, Infinite Sums
11. Integer Functions: Floors and Ceilings, Floor/Ceiling Applications, Floor/Ceiling Recurrences, 'mod': The Binary Operation, Floor/Ceiling Sums
12. Binomial Coefficients: Basic Identities, Basic Practice, Tricks of the Trade
13. Generating Functions
14. Hypergeometric Functions
15. Hypergeometric Transformations
16. Partial Hypergeometric Sums

Recommended Texts

1. Graham, R. L., Knuth, D. E., Patashnik, O. (1994). *Concrete mathematics: a foundation for computer science*. (2nd ed.). Boston: Addison-Wesley.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Awodey, S. (2010). *Category theory*. Oxford: Oxford University Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8422 **Number Theory and Cryptography** **3 (3+0)**

In this course the focus will be on the mathematical background that makes modern cryptography work as well as the mathematics behind cryptanalysis. The course includes basic number theory concepts, quadratic residues, public key cryptosystems, modern cryptosystems and elliptic curve cryptosystems. The concept of public key cryptography, advanced mathematical concepts, basics of number theory: time estimates for doing arithmetic, divisibility and the euclidean algorithm, congruences, some applications to factoring, finite fields and quadratic residues: finite fields quadratic residues and reciprocity, cryptography: some simple cryptosystems, enciphering matrices, public key: the idea of public key cryptography, rsa, discrete log, knapsack, zero-knowledge protocols, oblivious transfer, primality and factoring, pseudo primes, the rho method, fermat factorization and factor bases, the continued fraction method. Also, this course includes Elliptic Curves, Basics Elliptic curve cryptosystems, Elliptic curve primality test and Elliptic curve factorization. At the end of this course student able to develop or modify existing crypto algorithms.

Contents

1. The concept of public key cryptography
2. Advanced mathematical concepts
3. Basics of Number Theory: Time estimates for doing arithmetic
4. Divisibility and the Euclidean algorithm
5. Congruences, some applications to factoring
6. Finite Fields and Quadratic Residues: Finite fields
7. Quadratic residues and reciprocity
8. Cryptography: Some simple cryptosystems
9. Enciphering matrices
10. Public Key: The idea of public key cryptography
11. RSA
12. Discrete log
13. Knapsack
14. Zero-knowledge protocols
15. Oblivious transfer
16. Primality and Factoring
17. Fermat factorization and factor bases
18. The continued fraction method
19. The quadratic sieve method

Recommended Texts

1. Koblitz, N. (1994). *A course in number theory and cryptography*. New York: Springer-Verlag.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

3. Awodey, S. (2010). *Category theory*. Oxford: Oxford University Press.
4. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8423 **Harmonic Analysis** **3 (3+0)**

This course provides concepts for harmonic analysis using Riemann integral and metric spaces only. It provides an introduction to Fourier analysis, leading up to the poisson summation formula. The Fourier series and the Fourier transform are studied as special cases of locally compact abelian groups. Students shall learn techniques used in harmonic analysis. Fourier analysis leading up to the poisson summation formula, generalization of the concepts of Fourier analysis in the context of locally compact abelian groups and non-commutative harmonic analysis. Fourier analysis, Fourier series, Hilbert spaces, the Fourier transform, distributions, lca groups, finite abelian groups, the scholar will able to learn lca groups, the dual group plancherel theorem noncommutative groups, matrix groups the representations of su (2), the peter-weyl theorem and the Heisenberg group. This course also includes a brief introduction to Fourier Analysis and Fourier Series Hilbert Spaces. Moreover, generalization of the concepts of Fourier analysis in the context of locally compact abelian groups and non-commutative harmonic analysis.

Contents

1. Fourier analysis leading up to the Poisson summation formula
2. Generalization of the concepts of Fourier analysis in the context of locally compact abelian groups and non-commutative harmonic analysis.
3. Fourier Analysis
4. Fourier Series
5. Hilbert Spaces
6. The Fourier Transform
7. Distributions
8. LCA Groups
9. Finite Abelian Groups
10. LCA Groups
11. The Dual Group
12. Plancherel Theorem
13. Noncommutative Groups
14. Matrix Groups
15. The Representations of SU (2)
16. The Peter-Weyl Theorem
17. The Heisenberg Group

Recommended Texts

1. Deitmar, A. (2005). *A first course in harmonic analysis*. (2nd ed.). New York: Springer-Verlag
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Ceccherini-Silberstein, T., Scarabotti, F., Tolli, F. (2018). *Discrete harmonic analysis: representations, number theory, expanders, and the Fourier transform*. Cambridge: Cambridge University Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8424 **Randomized Algorithms** **3 (3+0)**

Randomized algorithms are used to simplify complex models at the cost of accuracy. The course is aimed to develop students' basic understanding in design and analysis of randomized algorithms. This course will also covers graph algorithms: all-pairs shortest paths, the min-cut problem minimum spanning trees approximate counting: randomized approximation schemes, the dnf counting problem parallel and distributed algorithms: the pram model, sorting on a pram, maximal independent sets, perfect matchings online algorithms: the online paging problem, adversary models paging against an oblivious adversary, relating the adversaries, the adaptive online adversary, the k-server problem. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Various randomize algorithms will also be taught along with its analysis. After the course the student will have a strong background of randomize algorithms theory which has diverse applications in the areas of computer science, biology, chemistry, physics, sociology, and engineering.

Contents

1. Concepts of a probability theory required for the course and usage of these concepts for design of algorithms.
2. Introduction: A Min-Cut Algorithm, Las Vegas and Monte Carlo, Binary Planar Partitions, A Probabilistic Recurrence
3. Game-Theoretic Techniques: Game Tree Evaluation, The Minimax Principle, Randomness and Non-uniformity
4. Moments and Deviations: Occupancy Problems, The Markov and Chebyshev Inequalities, Randomized Selection, Two-Point
5. Tail Inequalities: The Chernoff Bound, Routing in a Parallel Computer, A Wiring Problem, Martingales
6. The Probabilistic: Maximum Satisfiability, Expanding, Oblivious, The Lovasz Local Lemma, The Method of Conditional
7. Markov Chains and Random Walks: A 2-SAT Example, Markov Chains, Random Walks on Graphs, Cover Times
8. Graph Connectivity, Expanders and Rapidly Mixing Random Walks, Probability Amplification by Random Walks on Expanders

Recommended Texts

1. Motwani, R., Raghavan, P. (1995). *Randomized algorithms*. Cambridge: Cambridge University Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Feller, W. (1968). *An Introduction to probability theory and its applications*. Hoboken: John Wiley.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8425 **Spectral Graph Theory** **3 (3+0)**

Algebraic graph theory deals with studies of graphs by using algebraic properties of associated matrices like adjacency matrices, incidence matrices etc. whereas spectral graph theory relates graphs' properties to the spectrum of the adjacency matrices or Laplace matrices. The course is aimed to develop students' understanding of spectral properties of graphs to deduce theorems. The course includes Concepts of a graph spectrum, coloring concepts of linear algebra with emphasis on eigenvalues, spectra of trees, relationships between groups and graphs, Euclidean representations, Strongly regular graphs, Regular two-graphs, Association schemes, Distance regular graphs, p-ranks, Spectral characterizations, Graphs with few eigenvalues. Also, this course includes Trees: Characteristic polynomials of trees, Eigenvectors and multiplicities, Sign patterns of eigenvectors of graphs, Sign patterns of eigenvectors of trees, spectral center of a tree, Integral trees, (G,H,S), Spectrum, Non abelian Cayley graphs, Covers, Cayley sum graphs Topology: Embeddings, Minors, The Colin de Verdi`ere invariant, The Van der Holst-Laurent-Schrijver invariant, Euclidean representations: Examples, Euclidean representation, Root lattices, and Cameron-Goethals-Seidel-Shult.

Contents

1. Graph spectrum: Matrices associated to a graph, The spectrum of a graph, The spectrum of an undirected graph, Spectrum of some graphs, Decompositions, Automorphisms, Algebraic connectivity, Cospectral graphs
2. Linear algebra: Simultaneous diagonalization, Perron-Frobenius Theory, Equitable partitions, The Rayleigh quotient, Interlacing, Schur's inequality, Schur complements, The Courant-Weyl inequalities, Gram matrices, Diagonally dominant matrices
3. Eigenvalues and eigenvectors: The largest eigenvalue, Interlacing, Regular graphs, Bipartite graphs, Cliques and cocliques, Chromatic number, Shannon capacity, Classification of integral cubic graphs, The largest Laplace eigenvalue, Laplace eigenvalues and degrees, The Grone-Merris Conjecture, The Laplacian for hypergraphs, Applications of eigenvectors
4. The second largest eigenvalue: Bounds for the second largest eigenvalue, Large regular subgraphs are connected, Randomness, Expansion, Toughness and Hamiltonicity, Diameter bound, Separation, Block designs, Polarities

Recommended Texts

1. Brouwer, A.E., Haemers, W.H. (2012). *Spectra of graphs*, (1st ed.). Basel: Springer-Verlag.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Feller, W. (1968). *An Introduction to probability theory and its applications*. Hoboken: John Wiley.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8426 **Wavelets** **3 (3+0)**

Wavelets are useful data analysis tool in the time-frequency domain. They help to identify patterns in the data which are not visible in the time domain. Unlike fourier analysis, wavelets preserve time information along with frequency. This course shall aim to teach students some fundamentals of frequency domain analysis. This course aims to develop students' understanding in the design of wavelet filters for analysis of data so that they are able to do data analysis in the time-frequency domain. This course will introduce different basic level of transforms such as discrete Fourier transform, wavelets construction in $\mathbb{Z}$ and FFT, and introduction to complex numbers, linear algebra basics, basic properties of the discrete Fourier transform, translation-invariant linear transformations, fast Fourier transform. Further this course highlights the wavelets on $\mathbb{R}$: $l^2(\mathbb{R})$, approximate identities, the Fourier transform on $\mathbb{R}$, multiresolution analysis and wavelets construction of multiresolution analyses. This course includes implementation and a term paper.

Contents

1. Basics of complex numbers and linear algebra
2. Discrete Fourier transform
3. Wavelets construction in $\mathbb{Z}$.
4. Introduction to Complex Numbers
5. Linear Algebra basics
6. Basic Properties of the Discrete Fourier Transform
7. Translation-Invariant Linear Transformations
8. Fast Fourier Transform
9. Wavelets on $\mathbb{Z}_N$: Construction of Wavelets on $\mathbb{Z}_N$
10. First Stage and The Iteration Step
11. Examples and Applications
12. Wavelets on $\mathbb{Z}$: $l^2(\mathbb{Z})$
13. Complete Orthonormal Sets in Hilbert Spaces, $l^2([-\pi, \pi])$ and Fourier Series
14. The Fourier Transform and Convolution on $l^2(\mathbb{Z})$
15. First-Stage Wavelets on $\mathbb{Z}$, The Iteration Step for Wavelets on $\mathbb{Z}$, Implementation and Examples
16. Wavelets on $\mathbb{R}$: $l^2(\mathbb{R})$
17. Approximate Identities
18. The Fourier Transform on $\mathbb{R}$
19. Multiresolution Analysis and Wavelets
20. Construction of Multiresolution Analyses, Wavelets with Compact Support and Their Computation
21. Applications

Recommended Texts

1. Frazier, M. W. (2001). *An Introduction to wavelets through linear algebra*. New York: Springer
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Kaiser, G. (1994). *A friendly guide to wavelets* (1st ed.). New York: Springer
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8427 **Information Theory I** **3 (3+0)**

Information theory deals with the techniques and algorithms development for assessing now much information is present in the data. Moreover, if we are getting data from various sources, how this data share information. Information theory helps us to find common patterns among data as well as within data. Therefore, we can transmit or store data efficiently. The course will introduce students with key concepts of Information Theory like entropy, mutual entropy, cross entropy etc. Information Theory is one of the few scientific fields fortunate enough to have an identifiable beginning - Claude Shannon's 1948 paper. The story of the evolution of how it progressed from a single theoretical paper to a broad field that has redefined our world is a fascinating one. It provides the opportunity to study the social, political, and technological interactions that have helped guide its development and define its trajectory, and gives us insight into how a new field evolves. The course will in-traduce students with key concepts of Information Theory.

Contents

1. Information Sources
2. Pair Processes
3. Block Independent Channels
4. Ergodic Theorem
5. Entropy and The Entropy Ergodicity
6. Distortion and Approximation
7. Approximating Random Vectors and Processes
8. Distortion and Entropy
9. Relative Entropy
10. Information Rates
11. Distortion and Information
12. Ergodic Theorems for Densities
13. Source Coding Theorems
14. Asynchronous Block Codes
15. Sliding-Block Source Codes
16. Geometric Interpretation
17. Properties of Good Source Codes
18. Optimal Codes
19. Asymptotically Optimal Codes
20. Block Codes Sliding-Block Codes

Recommended Texts

1. Cover, T. M., Thomas, J. A. (2006). *Elements of information theory*. (2nd ed.). Hoboken: Wiley-Interscience.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Avery, J. S. (2012). *Information theory and evolution*, (2nd ed.). Singapore: World Scientific Publishing Company.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8428 **Special Topics in Big Data Analysis** **3(3+0)**

At the end of this course, the student will become familiar with the fundamental concepts of big data management and analytics; will become competent in recognizing challenges faced by applications dealing with very large volumes of data as well as in proposing scalable solutions for them; and will be able to understand how big data impacts business intelligence, scientific discovery, and our day-to-day activities and patterns. Performance analysis of serial and large-scale algorithms, employing hadoop map reduce, defining a big data strategy for your organization, enabling analytic innovation, implementing a big data solution, special topics in big data comparison of spark and with hadoop, comparison of flink with spark topics are also included. This course also includes delivering business benefit from big data, storing big data, analyzing your data characteristics, overview of big data stores, selecting big data stores, processing big data, integrating disparate data stores, employing hadoop mapreduce. The building blocks of hadoop mapreduce. Handling streaming data, tools and techniques to analyze big data.

Contents

1. Introduction to Big Data, Defining Big Data.
2. Delivering Business benefit from Big Data, Storing Big Data.
3. Analyzing your data characteristics.
4. Overview of Big Data stores, Selecting Big Data Stores.
5. Processing Big Data, Integrating disparate data stores.
6. Employing Hadoop MapReduce.
7. The building blocks of Hadoop MapReduce.
8. Handling streaming data, Tools and Techniques to Analyze Big Data.
9. Abstracting Hadoop MapReduce.
10. Performing ad hoc Big Data querying with Hive.
11. Creating business value from extracted data.
12. Developing a Big Data Strategy.
13. Series of Algorithms on Big Data
14. How large-scale Algorithms developed
15. Performance Analysis of serial and large-scale algorithms
16. Employing Hadoop MapReduce.

Recommended Texts

1. Mayer-Schönberger, V., & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton: Mifflin Harcourt.
2. H. Karau, A. Konwinski, P. Wendell, & Matei Zaharia. (2015). *Learning spark.* California: O'Reilly Media.

Suggested Readings

1. Rajaraman, A., & Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8429 **Advance topics in Fog Computing** **3(3+0)**

The primary purpose of this course is to capture the state-of-the-art in fog and edge computing, their applications, architectures, and technologies. In this course, researchers will identify and address various challenges in Fog Computing and develop suitable conceptual and technological solutions for tackling them. These include the development of scalable architectures, moving from closed systems to open systems, dealing with resource management, privacy and ethical issues involved in data sensing, storage, processing, and actions, designing interaction protocols, and autonomic management in various domains of Fog Computing from smart home, smart cities, science, industry, business, to consumer applications etc. This course will serve as a reference point for postgraduate students. This course is designed to give a multitude of technologies that comprise the modern concept of Fog and Edge computing, their applications, architectures, and technologies. Fog Computing is an emergent architecture for computing, storage, control, and networking that distributes these services closer to end users along the cloud-to-things continuum.

Contents

1. Internet of Things (IoT) and New Computing Paradigms
2. Fog and Edge Computing Principles, Architectures
3. Addressing the Challenges in Federating Edge Resources
4. Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds
5. Optimization Problems in Fog and Edge Computing
6. Data Management in Fog Computing
7. Predictive Analysis to Support Fog Application Deployment
8. Fog Computing Realization for Big Data Analytics
9. Machine Learning for Protecting the Security and Privacy of Internet of Things (IoT) Systems
10. Security, Resilience issues in Fog
11. Application Scenarios: Automation, Health care, Smart Cities, and Infotainment, streaming
12. Testing Perspectives of Fog-Based IoT Applications
13. Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit

Recommended Texts

1. Buyya, R., & Srirama, S. N. (Eds.). (2019). Fog and edge computing: principles and paradigms.
2. Hoboken: Wiley.
3. 2. Abbas, A., Khan, S. U., & Zomaya, A. Y. (Eds.). (2020). Fog computing: Theory and Practice.
4. Hoboken: Wiley.

Suggested Readings

1. 1. Prabhu, C. S. R. (2019). Fog computing, deep learning and big data analytics-research
2. Directions (1st ed.). Singapore: Springer
3. 2. Rahmani, A. M., Liljeberg, P., Preden, J. S., & Jantsch, A. (Eds.). (2017). Fog computing in the internet of things: Intelligence at the edge. Cham: Springer.

CSEC-8430 Advance Topics in Cloud Computing 3(3+0)

The primary aim of this course is to provide students with a proper understanding of cloud computing, its risks, how to implement and manage an effective cloud strategy. The course also provides case studies of numerous existing computes, storage, and application cloud services and illustrates the capabilities and limitations of current providers of cloud computing services. This allows the students to understand the mechanisms needed to harness cloud computing in their own respective endeavors. In this course, many open research problems that have arisen from the rapid uptake of cloud computing are detailed. Finally, graduates of this course are capable of modelling and simulation of large-scale Cloud computing environments by CloudSim for testing Cloud Computing-related hypotheses to solve a certain set of problem(s). In addition, this course includes the study of recent research papers. Also, this course includes a term paper with extensive literature review of the assigned topic with critical analysis.

Contents

1. Cloud Computing Principals, Architectures and Applications
2. Web Services delivered by Cloud
3. Cloud-Enabling Technology, Virtualization, Server consolidation
4. Cloud Infrastructure Mechanisms
5. Specialized Cloud Mechanisms
6. Resource Management in Cloud
7. Scheduling in Cloud
8. Reliability, Security, Privacy, Robustness, challenges and solutions
9. Data Management in Cloud Computing
10. Modeling and Simulation of Cloud Computing Environments using CloudSim
11. Challenges for multi-objective optimization of resources in cloud

Recommended Texts

1. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. New York: CRC press.
2. Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2010). *Cloud computing: principles and paradigms* (Vol. 87). Hoboken: Wiley.

Suggested Readings

1. Erl, T., Puttini, R., & Mahmood, Z. (2013). *Cloud computing: concepts, technology & architecture*. New York: Pearson Education.
2. Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering cloud computing: foundations and applications programming*. Waltham, MA: Elsevier.
3. Buyya, R., & Dastjerdi, A. V. (Eds.). (2016). *Internet of Things: Principles and paradigms*. Cambridge, MA: Elsevier.

CSEC-8431 Natural Language Engineering 3 (3+0)

In this course we will explore fundamentals of natural language processing. Natural language processing (NLP) or computational linguistics is one of the most important technologies of the information age. Applications of NLP are everywhere because people communicate almost everything in language: web search, advertising, emails, customer service, language translation, virtual agents, medical reports, etc. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning.

Contents

1. Introduction, Chomsky hierarchy, Language models.
2. Probability concepts, Bayes' Theorem, Smoothing n-grams.
3. Improving CFG with attributes, Context-free parsing, Earley algorithm, Extending CFG.
4. Probabilistic parsing, Parsing tricks, Human sentence processing.
5. Semantics, Forward-backward algorithm, Expectation Maximization.
6. Finite-state algebra, Finite-state implementation, Finite-state tagging, Noisy channels and FSTs, More FST examples.
7. Programming with regexps, Morphology and phonology.
8. Optimal paths in graphs, Structured prediction.
9. Current NLP tasks and competitions, Applied NLP, Topic models, Machine translation.

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing*. London: Prentice Hall.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8432 Advance Natural Language Engineering 3 (3+0)

In this course we will explore statistical, model-based approaches to natural language processing. There will be a focus on corpus-driven methods that make use of supervised and unsupervised machine learning approaches and algorithms. We will examine some of the core tasks in natural language processing, starting with simple word-based models for text classification and building up to rich, structured models for syntactic parsing and machine translation. In each case we will discuss recent research progress in the area and how to design efficient systems for practical user applications. We will also introduce the underlying theory from probability, statistics, and machine learning that are crucial for the field, and cover fundamental algorithms like n-gram language modelling, naive Bayes and Maxent classifiers, sequence models like Hidden Markov Models, probabilistic dependency and constituent parsing, and vector-space models of meaning. In this course, students will gain a basic introduction to cutting-edge research in Deep Learning for NLP. Through lectures, assignments and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models.

Contents

1. Text Classification via Naive Bayes & Maximum Entropy, Classification Tutorial, MaxEnt Tutorial, Generative and Discriminative Classifiers.
2. Hidden Markov Models, Clustering, Part-of-Speech Tagging, TnT Tagger.
3. Advanced Part-of-Speech Tagging, Word Alignments, MT Tutorial, Overview, IBM Models, HMM-Alignments, Agreement.
4. Phrase-Based Translation, Decoding, Phrases
5. Syntactic Parsing, Best-First, A*, K-Best, Shift-Reduce Parsing.
6. Advanced Constituency Parsing, Unlexicalized, Lexicalized, Latent Variable.
7. Semantic Parsing, Hierarchical (Syntax-Based) Translation, Hiero, GHKM, Syntax vs. Phrases, Synchronous Grammars.
8. Sentiment Analysis, Aspects, Lexicons, Summarization.
9. Summarization, Query, N-Gram, Topical.
10. Lexical Acquisition.

Recommended Texts

1. Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing*. London: Prentice Hall.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. California: OReilly Media.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

CSEC-8433 **Fundamentals of Text Processing** **3 (3+0)**

The processing and analysis of large datasets has become a regular task in sciences. This introductory course into the scripting language PERL provides the basis for designing rapid, reproducible and scalable solutions to this problem. The scripting language PERL is an intuitive and powerful tool for developing custom-tailored solutions for problems ranging from basic data handling and management up to the design of complex workflows and novel algorithms for data analysis. In this course we will introduce the basic concepts of PERL, making you familiar with the various data types and the general structure of PERL scripts, but also with the basic concepts of a structured and standardized data analysis. Based on specific examples from NLP we will guide you through the implementation of first algorithms in PERL aiding in the solution of your particular data analysis problems. In this course, you'll learn natural language processing (NLP) basics, such as how to identify and separate words, how to extract topics in a text. This course will give you the foundation to process and parse text as you move forward in your PERL learning.

Contents

1. Background, Introduction to Perl.
2. Scalar Data, Built in Functions.
3. Arrays, Functions, Writing Safe Code.
4. Control Structures, File Input / Output.
5. Introduction to Text Processing, Text Processing Functions.
6. Loop Control, Hashes, DBM Databases, Advanced Sorting.
7. Regular Expressions, Environment Variables, CGI-Programming.
8. Process Management, References and Data Structures.
9. Graphics, Javascript

Recommended Texts

1. Schwartz, R. L., & Phoenix, T. (2001). *Learning perl*. California: O'Reilly Media.
2. Christiansen, T., Wall, L., & Orwant, J. (2012). *Programming perl: unmatched power for text processing and scripting*. California: O'Reilly Media.

Suggested Readings

1. Christiansen, T., & Torkington, N. (2003). *Perl cookbook: solutions & examples for pearl programmers*. California :O'Reilly Media.
2. Lidie, S., & Walsh, N. (2002). *Mastering perl: graphical user interfaces in perl*. California: O'Reilly Media.

CSEC-8434 Fundamentals of Speech Recognition 3 (3+0)

This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically extracting text from human speech. This class will cover many theoretical and practical aspects of machine learning techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition. Fundamentals of Speech Recognition, is a comprehensive course, covering all aspects of automatic speech recognition from theory to practice. In this course such topics as Anatomy of Speech, Signal Representation, Phonetics and Phonology, Signal Processing and Feature Extraction, Probability Theory and Statistics, Information Theory, Metrics and Divergences, Decision Theory, Parameter Estimation, Clustering and Learning, Transformation, Hidden Markov Modelling, Language Modelling, Neural Networks (specifically TDNN, LSTM, RNN, and CNN architectures) plus other recent machine learning techniques used in speech recognition are covered in some detail.

Contents

1. Overview of Course, Intro to Probability Theory, and ASR Background: N-gram Language Modeling
2. TTS: Background (part of speech tagging, machine learning, classification, NLP) and Text Normalization, Phonetics Speech Synthesis, pages 1-10 , Optional Advanced Reading, Text Segmentation and Organisation, Text Decoding.
3. TTS: Grapheme-to-phoneme, Prosody (Intonation, Boundaries, and Duration) and the Festival software, Prosody Prediction from Text.
4. TTS: Waveform Synthesis (Diphone and Unit Selection Synthesis), Unit Selection Synthesis, Optional Advanced Reading.
5. ASR: Noisy Channel Model, Bayes, HMMs, Forward, Viterbi, Hidden Markov Models, Automatic Speech Recognition.
6. ASR: Feature Extraction and Acoustic Modeling, Evaluation, Speech Recognition: Advanced Topics.
7. ASR: Learning (Baum-Welch) and Disfluencies, Automatic Speech Recognition, Speech Recognition: Advanced Topics.

Recommended Texts

1. Jurafsky, D. Martin, J. H. (2019). *Speech and language processing* (3rd ed.). New Delhi: Pearson Education India.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.

Suggested Readings

1. Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge: Cambridge university press.
2. Jones, M. T. (2008). *Artificial intelligence: a systems approach*. New Delhi: Laxmi Publications.