UNIVERSITY OF SARGODHA
OFFICE OF THE REGISTRAR
(ACAD BRANCH)

NOTIFICATION

On the recommendations of Academic Council made in its 21st (2/2024) meeting held on 07.06.2024, the Syndicate in its 67th (3/2024) meeting held on 12.07.2024 approved the following for implementation w.e.f. Fall 2024 at Main Campus and Affiliated Colleges:

i. Curriculum of Associate Degree in Computer Science (Annex-‘A’)

ii. Revised curriculum of BS in Computer Science (Annex-‘B’)

iii. Curriculum of BS in Computer Science (5th Semester Intake) (Annex-‘C’)

iv. Revised curriculum of Associate Degree in Information Technology (Annex-‘D’)

v. Revised curriculum of BS in Information Technology (Annex-‘E’)

vi. Revised curriculum of BS in Information Technology (5th Semester Intake) (Annex-‘F’)

vii. Curriculum of Associate Degree in Software Engineering (Annex-‘G’)

viii. Revised curriculum of BS in Software Engineering (Annex-‘H’)

ix. Curriculum of BS in Software Engineering (5th Semester Intake) (Annex-‘I’)

(WAQAR AHMAD)
Additional Registrar (General)

No. SU/Acad/24/747

Dated: 26.09.2024

Distribution:

- Chairman, Department of Computer Science
- Chairman, Department of Information Technology
- Chairman, Department of Software Engineering
- Controller of Examinations
- Director Academics

C.C:

- Dean, Faculty of Computing & Information Technology
- Director, QEC
- Deputy Registrar (Affiliation)
- Deputy Registrar (Registration)
- Secretary to the Vice-Chancellor
- PA to Registrar
- Notification File

Annex-'G'

Curriculum

of

Associate Degree in Software Engineering

for

Main Campus and Affiliated Colleges

Department of Software Engineering
University of Sargodha

(Applicable from Fall 2024)

Table of Contents

Vision Statement of University of Sargodha

Our vision is to contribute to society through the pursuit of quality education, creative learning and productive research

Mission Statement of University of Sargodha

Our mission is:

- To provide students with a healthy learning experience based on critical thinking, innovation and leadership skills
- To ensure a collaborative work environment for faculty and staff to achieve professional excellence and institutional growth
- To contribute to knowledge economy and social transformation through advanced studies and research

Vision Statement of Department of Software Engineering

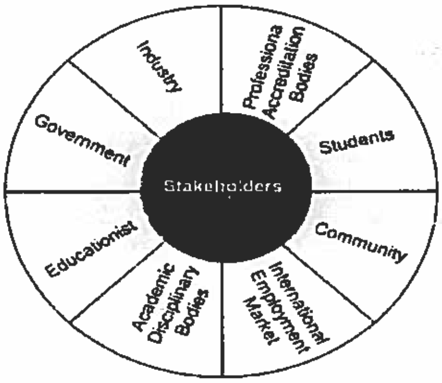
Our vision is to enable sustainable development of society by providing quality education, creative learning, and productive research in the field of software engineering.

Mission Statement of the Department of Software Engineering

Our mission is to:

- Prepare software engineering graduates to have strong knowledge, critical thinking, research, and leadership skills by providing a healthy learning environment.
- Bridge the gap and enhance the productive interaction with the national and international level academia and software industry for mutual benefits.
- Contribute to the knowledge economy, social transformation, and community services through advanced software studies andinterdisciplinary research.

Underlying Principles of SE Degree Programs


Industry
Professiona Accreditation Bodies
Students
Community
International Employment Market
Academic Disciplinary Bodies
Educationist
Government
Stakeholders

Curriculum plays an important role within education as it outline the planned and structured learning experiences that an academic program provides. For an effective academic program, the curriculum must meet the needs of the stakeholders and face the emerging challenges. The Department of Software Engineering (UOS) realizes the rapidly changing needs of today's knowledge intensive technology driven complex work places and the changing patterns of 21st century universities' education which have removed the identity of place, the identity of time, the identity of the scholarly community, and the identity of the student community. To meet these challenges, the Department has revised the existing curriculum. The curriculum is based on following underlying principles:

i. The curriculum should be a broad based and provides students with the flexibility to work across many disciplines & professions.
ii. The curriculum should prepare graduates to succeed in a rapidly changing field.
iii. The curriculum should provide guidance for the expected level of mastery of topics by graduates.
iv. Should provide realistic, adoptable recommendations that provide guidance and flexibility, allowing curricular designs that are innovative and track recent developments in the field.
v. The curriculum contents should be relevant and compatible with a variety of institutions.
vi. The size of the essential knowledge must be managed.
vii. The curriculum should identify the fundamental skills and knowledge that all graduates should possess.
viii. The curriculum should provide the greatest flexibility in organizing topics into courses and curricula.

The curriculum has been developed using top-down curriculum development approach. It has adopted a balanced and multidisciplinary approach and presents a blend of study areas which spreads across the boundaries of fundamental knowledge of traditional disciplines to advanced knowledge of the emerging disciplines. Body of knowledge (BOK) of AI program covers knowledge areas which are required for the program's accreditation from the Accreditation Council and knowledge area which are required for professional certification and professional development.

It is universally accepted that each profession needs both a specific skill set and an appropriate mindset. Developing an appropriate mindset of the prospective computing graduates requires a body of knowledge which enriches students' experiences, thoughts, beliefs, assumptions, and attitudes about the special characteristics of that specific domain. Therefore, the course contents and related practical experiences are designed to meet the professional requirements of the respective domain. For this the revised curriculum mainly focuses on following six (6) key areas:

i. Knowledge: Theoretical learning of concepts and principles regarding a particular subject(s).
ii. Skills: Capability of using learnt knowledge and applying it according to the context
iii. Competencies: The ability to do things satisfactory- not necessarily outstandingly or even well, but rather to a minimum level of acceptable performance.
iv. Expertise: Level of proficiency and innovative ways of applying learnt knowledge. (Competitive edge)
v. Dispositions: Habits of mind or tendencies to respond to certain situations in certain ways. The role of dispositions in computing education is very important. For example, having the disposition to be a programmer is much better that just having programming skills.
vi. Values: Moral, ethical and professional practices.

To strengthen the curriculum further, specialization tracks have also been integrated within the curriculum's BOK. These specialization tracks are designed according to what the industry is looking for in an employee and the learning interests of students. Furthermore, life skills including desired dispositions, soft skills, public speaking, critical thinking & reasoning, 21st Century literacies, personal attributes, entrepreneurship, attitude towards lifelong learning, professional practices and other social skills have not considered discrete items, rather threaded into the entire fabric of the curriculum.

Curriculum

For
Associate Degree in Software Engineering

Programs' Rationale

The 21st century is loaded with a large number of challenges. These challenges include globalized business environment, keeping pace with innovative technologies, the availability of information with respect to time, speed, volume, mode, nature and management of this exponentially growing information, keeping control on international and inter-organizational business processes in real time, optimization of business processes across multiple sites, highly uncertain and chaotic business environments, a new level of national & international competition (hyper-competition), social & cultural diversity, rapidly changing products and processes, government regulations, increasing importance of skills, qualities, productivity and other stresses. To face these challenges and to bring a high level of agility, control and transparency organizations now increasingly focus on maximizing their existing technology and human infrastructure through automating various processes that can free human resource to add value elsewhere within the organization. Accordingly, the software industry looks for graduates who are not only equipped with conventional computing skills but also have the capability to develop complex software that can provide verifiable insight into underlying business processes.

Software Engineering is the discipline of developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. Software engineering is different in character from other engineering disciplines due to both the intangible nature of software and the related operations. It seeks to integrate the principles of mathematics and computer science with the engineering practices developed for tangible, physical artifacts. Software engineering students learn more about software reliability and maintenance and focus more on developing and maintaining software techniques while Computer Science students just acquire abstract knowledge of these aspects.

Software Engineering is a bridge connecting the basic concepts and principles of Computer Science with a variety of users who can benefit from technologies based upon those principles. It includes the design and development of software systems which are effective, efficient, robust, maintainable, and maximally useful and usable. It also includes the design and development of techniques, processes and higher level tools by which these applications can be developed in a timely, cost effective and sustainable manner. At both levels, it requires a systematic approach which deals with quantifiable measures of quality and effectiveness, as well as attention to the critical nature of various products of the process. Software Engineering, therefore, requires familiarity with the basic needs and processes in the various application domains, with the principles of good engineering practices and with the underlying concepts and principles of computer science. It requires facility in problem analysis, solution design, program development and documentation. It also requires a basic understanding of ways in which humans interact with technological systems and necessary skills to create high-quality software systems in a systematic, controlled and efficient manner. It involves the application of engineering concepts, techniques, and methods to the design, development, deployment and maintenance of software systems.
The key rationale behind Associate Degree in Software Engineering is to produce graduates who have mastery in the above discussed aspects. The program intends to impart knowledge and training which enable students to harmonize a theory with practice, a concept with an application, and a problem with a solution. It will prepare them to apply ably engineering principles, practices, and processes to design, develop, deploy, and maintain software systems. The program will lead to development of students'

professional and interpersonal skills. It will help them to enhance their ability in oral and written communication, and their adaptability to team environments. The program will inculcate among students a strong sense of civic, professional and ethical responsibility. The program will also strive to develop a capacity for innovation and a passion for lifelong learning.

Curricula Consideration

During the revision of the Computing Curricula two major guidelines have been considered (ACM and Seoul Accord). However, in some cases the main focus of these guidelines is mostly traditional Software Engineering programs.

Association of Computing Machinery (ACM) - Guidelines

Association of Computing Machinery (ACM), USA is the largest body in the world for computer scientists (IT professionals). Its membership is spread over the entire globe. It has a pool of highly reputed professionals which meet after a few years to assess the directions being taken by the computing discipline. In view of its assessment, it identifies knowledge areas and also their relative importance in the years to come. Thus, ACM shows the path to follow to the computing academia and professionals all over the world. Computing curricula are designed keeping in view following identified knowledge areas of ACM [ref # ACM 2013 curriculum report]. It has been tried to reasonably cover all knowledge areas without compromising the flexibility needed for a national model curriculum. The mapping of these key knowledge areas with the courses are given in the table below.

- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visual Computing
- HCI - Human-Computer Interaction
- IAS - Information Assurance and Security
- IM - Information Management
- IS - Intelligent Systems
- NC - Networking and Communications
- OS - Operating Systems
- PBD - Platform-based Development
- PD - Parallel and Distributed Computing
- PL - Programming Languages
- SDF - Software Development Fundamentals
- SE - Software Engineering
- SF - Systems Fundamentals
- SP - Social Issues and Professional Issues

The following knowledge areas have been addressed with the major computing courses.

Knowledge Areas in ACM CS 2013 Curriculum

	Knowledge Area	CS 2013		ACM 2013 Subjects Taught in Various Universities	NCEAC Revised 2023 Subjects in Core
		Tier-1	Tier-2		
1	AL-Algorithms and Complexity	19	9	Algorithms; Algorithms and Data Structures; Algorithm Design and Analysis	Data structures, Analysis of Algorithms, Theory of Automata
2	AR-Architecture and Organization	0	16	Intro to Computer Architecture; DLD; Computer Engineering	DLD, Computer Org & Assembly Language, Computer Architecture
3	CN-Computational Science	1	0	eScience; Modeling and Simulation; Computer Graphics	HCI & Computer Graphics; (Elective: Numerical Analysis)
4	DS-Discrete Structures	37	4	Discrete Mathematics; Mathematical Foundations of CS; Probability for CS; Discrete Structures 1; Discrete Str 2	Discrete Structures, Probability & Statistics
5	GV-Graphics and Visualization	2	1	Computer Graphics; Computer Graphics	HCI & Computer Graphics; (Elective: Computer Graphics)
6	HC-Human-Computer Interaction	4	4	Human Computer Interaction	HCI & Computer Graphics
7	IAS-Information Assurance and Security	3	6	Computer Systems Security	Information Security; (Elective: Cyber Security)
8	IM-Information Management	1	9	Database Systems	Database Systems; Adv Database Management Sys
9	IS-Intelligent Systems	0	10	Artificial Intelligence Programming; Artificial Intelligence	Artificial Intelligence
10	NC-Networking and Communication	3	7	Introduction to Computer	Computer Networks

				Networking; Computer Networks	

Outcome Based Education (OBE) System and Seoul Accord:

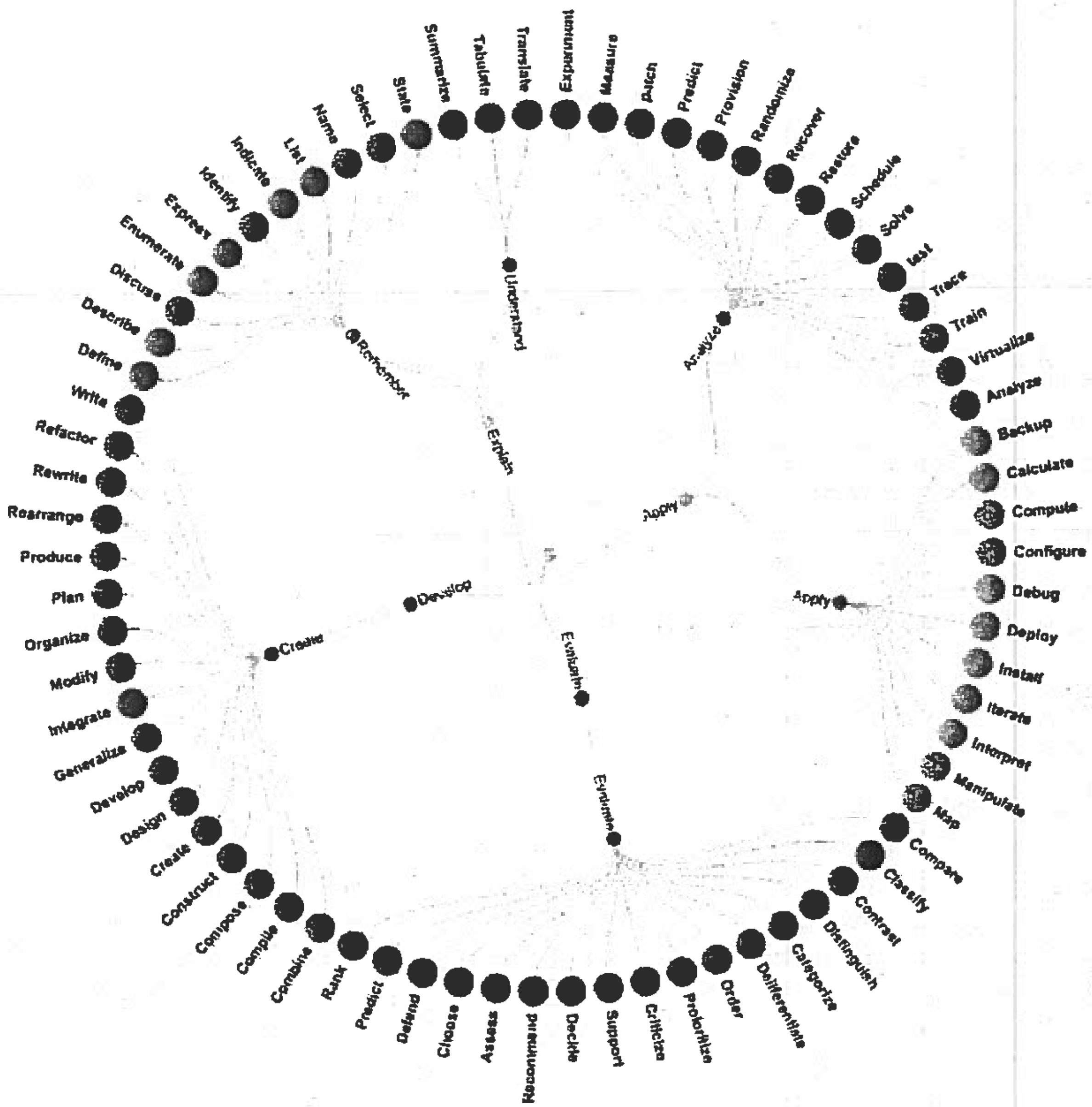
Keeping in view the latest transformation from knowledge-based education philosophy to Outcome based education (OBE) system, the OBE model based on Seoul Accord has also been considered. Computing programs prepare students to attain educational objectives by ensuring that students demonstrate achievement of the following outcomes (derived from Graduate Attributes define by Seoul Accordwww.seoulaccord.org).

S#	Program Learning Outcomes (PLOs)	Computing Professional Graduate
1	Academic Education	To prepare graduates as computing professionals
2	Knowledge for Solving Computing Problems	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.
3	Problem Analysis	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.
4	Design/ Development of Solutions	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
5	Modern Tool Usage	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
6	Individual and Team Work	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.

7	Communication	Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
8	Computing Professionalism and Society	Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice
9	Ethics	Understand and commit to professional ethics, responsibilities, and norms of professional computing practice
10	Life-long Learning	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional

Bloom's Taxonomy

Revised Bloom's Taxonomy	**Skill level with applicable verbs**
Remember	**Explain:** define, describe, discuss, enumerate, express, identify, indicate, list, name, select, state, summarize, tabulate, translate
Understand	
Apply	**Apply:** backup, calculate, compute, configure, debug, deploy, experiment, install, iterate, interpret, manipulate, map, measure, patch, predict, provision, randomize, recover, restore, schedule, solve, test, trace, train, virtualize
Analyze	**Evaluate:** analyze, compare, classify, contrast, distinguish, categorize, differentiate, discriminate, order, prioritize, criticize, support, decide, recommend, assess, choose, defend, predict, rank
Evaluate	
Create	**Develop:** combine, compile, compose, construct, create, design, develop, generalize, integrate, modify, organize, plan, produce, rearrange, rewrite, refactor, write



Program Educational Objectives (PEOs)

BSSE Program aims to create, expand, disseminate and teach the Software Engineering body of knowledge through academics, applications and research which positively impact society locally, nationally, and internationally.

The objective of the program is to prepare students for professional careers and graduate studies with a balance between computing theory and practical application of software engineering concepts, methodologies, tools and technologies in the modern software development environments. The curriculum is designed to ensure breadth across allied disciplines and supporting subjects; and depth in most areas of the software engineering body of knowledge. Various components have been included in the curriculum to ensure that the graduates:

PEO-1: Have the ability to effectively apply Software Engineering principles and best practices throughout different stages of Software Development Life Cycle (SDLC) to develop quality software applications.

PEO-2: Have an awareness of current industry standards and practices.

PEO-3: Are able to understand and apply software project management skills

PEO-4: Have strong communication, team management and interpersonal skills.

PEO-5: Understand professional responsibility and application of ethical principles.

PEO-6: Are able to translate their skills to knowledge economy and socio-economic growth of the country.

Program's Outcome

The program will produce entrepreneurs of great character, competence, vision and drive equipped with up-to-date knowledge, marketable skills, valuable competencies, unique expertise, globally compatible dispositions and culturally and professionally acceptable values to take on appropriate professional roles in software engineering domain or proceed to further or higher education or training.

Program's Structure

1. Associate Degree in Software Engineering comprises of 04 semesters / 02 years duration.
2. Associate Degree in Software Engineering may only be offered under semester system.
3. Semester duration is 18 weeks, including two weeks for examinations and results
4. The minimum credits for award of Associate Degree Computing is 72 Credit Hours
5. Normal load per semester will be five to six courses and maximum up to 8 courses under special circumstances as per the university/DAI rules and regulations.
6. Courses will be described through Credit Hours (CrHr) system.
7. 1 (one) Theory CrHr is equivalent to 1 (one) contact hour per week in a normal semester of 15/16 weeks teaching so a 3 CrHr course means 45 contact hours for the whole semester.
8. 1 Lab CrHr is equivalent to 3 contact hours per week for 15/16 weeks

Eligibility Criteria

The minimum requirements for admission in a bachelor degree program in Software Engineering is at least 50% marks in Intermediate (HSSC) examination with one of the following combinations:

1. Pre-Engineering
2. Pre-Medical (Admitted candidates have to pass 6-credit hours courses of mathematics in first two semesters.)
3. General Science
   a. Mathematics, Statistics, Physics
   b. Mathematics, Statistics, Economics

c. Mathematics, Statistics, Computer
d. Mathematics, Physics, Computer
e. Mathematics, Economics, Computer

4. A-Levels (with equivalence of mentioned above by IBCC) with at-least 50% obtained marks

Degree Requirement

To become eligible for award of Associate Degree degree, a student must satisfy the following requirements:

a) Must have studied and passed the prescribed courses, totaling at least 72 credit hours.
b) Must have earned CGPA (Cumulative Grade Point Average) of at least 2.0 on a scale of 4.0.

Duration

Associate Degree in Software Engineering comprises of 04 semesters / 02 years duration.

Degree Title

Degree title will be "Associate Degree in Software Engineering".

Assessment & Evaluation

University's semester and examination rules & regulations shall be followed for assessment & evaluation.

Curriculum AD in Software Engineering

Studentswillberequiredtocompletethe followingcoursestoobtainAssociateDegreeCS.

GenericStructureforComputingDisciplines:

Areas	**CreditHours**	**Courses**
ComputingCore	34	10
Elective	14	5
Mathematics&SupportingCourses	3	1
GeneralEducation Requirement	21	8
Totals	**72**	**24**

Mapping of ADCS Program on the Generic Structure:

#	Sem #	Code	Pre-Reqs	Course Title	Dom	CrHr
Computing Core (34/72) 10 Courses						
1	1	CMPC-5201		Programming Fundamentals	Core	4(3-3)
2	2	CMPC-5202	CMPC-5201	ObjectOriented Programming	Core	4(3-3)
3	2	CMPC-5203		Database Systems	Core	4(3-3)
4	2	CMPC-5204		DigitalLogic Design	Core	3(2-3)
5	3	CMPC-5205	CMPC-	Data Structures	Core	4(3-3)

			5202			
6	3	CMPC-5206		Information Security	Core	3(2-3)
7	3	CMPC-5207		Artificial Intelligence	Core	3(2-3)
8	3	CMPC-5208		Computer Networks	Core	3(2-3)
9	5	CMPC-5209	CMPC-5204	ComputerOrganization&Assembly Language	Core	3(2-3)
10	4	CMPC-5101		Software Engineering	Core	3(3-0)
Electives (14/72) 5 Courses (Universities can add more Lab Courses)						
11		SEDE-2201		Object Oriented Analysis & Design Lab	Elective	2(0-6)
12		CSDE-5202		WebTechnologies Lab	Elective	3(1-6)
13		SEDE-2202		MobileApplicationDevelopment Lab	Elective	3(1-6)
14		CSDE-2203		AdvancedProgramming Lab	Elective	3(1-6)
15		SEDE-2203		Software Verification and Validation Lab	Elective	3(1-6)
Mathematics Supporting Courses (3/72) 1 Course						
16	2	MATH-5102	URCQ-5102	Linear Algebra	Maths	3(3-0)
General Education Requirement (21/72) 8 Courses						
17	1	URCS-5123		ApplicationofInformation&Communication Technologies	GER	3(2-3)
18	1	URCE-5118		Functional English	GER	3(3-0)
19	2	URCE-5119		Expository Writing	GER	3(3-0)
20	1	URCQ-5102		QuantitativeReasoning–1(Discrete Structures)	GER	3(3-0)
21	1	URCQ-5103		QuantitativeReasoning–2(CalculusandAnalytic Geometry)	GER	3(3-0)
22	4	URCI-5105		Islamic Studies	GER	2(2-0)
23	8	URCI-5122		IdeologyandConstitutionof Pakistan	GER	2(2-0)
24	4	URCE-5124		Entrepreneurship	GER	2(2-0)

Semester/StudyPlanfor ADSE

#	Code	Pre-Reqs	Course Title	Domain	Cr Hr (Cont hr)
			Semester 1		
1	CMPC-5201		ProgrammingFundamentals	Core	4(3-3)
2	URCS-5123		ApplicationofInformation&CommunicationTechnologies	GER	3(2-3)
3	URCQ-5102		QR1(DiscreteStructures)	GER	3(3-0)
4	URCQ-5103		QR2(CalculusandAnalyticGeometry)	GER	3(3-0)
5	URCE-5118		FunctionalEnglish	GER	3(3-0)
	6101			**Total Cr Hrs**	**16 (14-6)**
			Semester 2		
6	CMPC-5202		ObjectOrientedProgramming	Core	4(3-3)
7	CMPC-5203		DatabaseSystems	Core	4(3-3)
8	CMPC-5204		DigitalLogicDesign	Core	3(2-3)
9	MATH-5102		LinearAlgebra	Maths	3(3-0)
10	URCE-5119		ExpositoryWriting	GER	3(3-0)
				TotalCr Hrs	**17(14-9)**
			Semester 3		
11	CMPC-5205		Data Structures	Core	4(3-3)
12	CMPC-5206		InformationSecurity	Core	3(2-3)
13	CMPC-5207		ArtificialIntelligence	Core	3(2-3)
14	CMPC-5208		ComputerNetworks	Core	3(2-3)
15	CMPC-5101		SoftwareEngineering	Core	3(3-0)
16	CMPC-5209		ComputerOrganization&AssemblyLanguage	Core	3(2-3) New course ?
				Total Cr Hrs	**19 (14-15)**
			Semester 4		
17	SEDE-2201		Elective1(Example:Object Oriented Analysis & Design Lab)	Elective	2(0-6)
18	CSDE-5202		Elective2(Example:WebTechnologiesLab)	Elective	3(1-6)
19	CSDE-2202		Elective3(Example:MobileApplicationDevelopmentLab)	Elective	3(1-6)
20	CSDE-2203		Elective4(Example:AdvancedProgrammingLab)	Elective	3(1-6)
21	SEDE-2203		Elective5(Example:Software Verification and Validation Lab)	Elective	3(1-6)

Credit ML 11

Miss Cours
5101
5111

Miss Course
5103

?

?

22	URCI-5105		IslamicStudies	GER	2(2-0)
23	URCI-5122		IdeologyandConstitutionofPakistan	GER	2(2-0)
24	URCE-5124		Entrepreneurship	GER	2(2-0)
				Total Cr Hrs	**20 (10-30)**

Course Contents

CMPC-5201 Programming Fundamentals

Credit Hours: **Contact Hours:** **Pre-requisites:**	4 (3-3) 3-3 None

Course Introduction:

This course provides fundamental concepts of programming to freshmen. The courses is prerequisite
to many other courses, therefore, students are strongly advised to cover all contents and try to achieve CLOs to the maximum possible level. The course may be taught as language independent. Further, it is up to the university to choose any language for the practical/Lab purpose but that must be latest and market oriented.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand basic problem solving steps and logic constructs	C2 (Understand)
CLO-2	Apply basic programing concepts	C3 (Apply)
CLO-3	Design and implement algorithms to solve real world problems	C3 (Solve)

Course Outline:

Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, multidimensional lists, introduction to modular programming, function definition and calling, stack rolling and unrolling, string and string operations, pointers/references, static and dynamic memory allocation, File I/O operations.

Reference Materials (or use any other standard and latest books):

1. Starting out with Programming Logic & Degins, 4th Edition, Tony Gaddis,
2. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie
3. Object Oriented Programming in C++ by Robert Lafore

4. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel
5. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

URCS-5123 Application of Information & Communication Technologies

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	None

Course Introduction:

This is an introductory course in Computer Science designed for beginners. Apart from leading the participants through a whirlwind history of computing, the course also develops a feel for web programming through a series of lectures that help the students develop their own web page. Main objective of the course is to build an appreciation for the fundamental concepts in computing and to become familiar with popular PC productivity software.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand basics of computing technology	C1 (Knowledge)
CLO-2	Do number systems conversions and arithmetic	C2 (Understand)
CLO-3	Have knowledge of types of software	C2 (Understand)
CLO-4	Have knowledge of computing related technologies	C3 (Apply)

Course Outline:

Brief history of Computer, Four Stages of History, Computer Elements, Processor, Memory, Hardware, Software, Application Software its uses and Limitations, System Software its Importance and its Types, Types of Computer (Super, Mainframe, Mini and Micro Computer), Introduction to CBIS (Computer Based Information System), Methods of Input and Processing, Class2. Organizing Computer Facility, Centralized Computing Facility, Distributed Computing Facility, Decentralized Computing Facility, Input Devices. Keyboard and its Types, Terminal (Dump, Smart, Intelligent), Dedicated Data Entry, SDA (Source Data Automation), Pointing Devices, Voice Input, Output Devices. Soft- Hard Copies, Monitors and its Types, Printers and its Types, Plotters, Computer Virus and its Forms, Storage Units, Primary and Secondary Memories, RAM and its Types, Cache, Hard Disks, Working of Hard Disk, Diskettes, RAID, Optical Disk Storages (DVD, CD ROM), Magnetic Types, Backup System, Data Communications, Data Communication Model, Data Transmission, Digital and Analog Transmission, Modems, Asynchronous and Synchronous Transmission, Simplex. Half Duplex, Full Duplex Transmission, Communications, Medias (Cables, Wireless), Protocols, Network Topologies (Star, Bus, Ring), LAN, LAN, Internet, A Brief History, Birthplace of ARPA Net, Web Link, Browser, Internet Services provider and Online Services Providers, Function and Features of Browser, Search Engines, Some Common Services available on Internet.

Reference Materials (or use any other standard and latest books):

1. Charles S. Parker, Understanding Computers: Today and Tomorrow, Course Technology, 25 Thomson Place, Boston, Massachusetts 02210, USA
2. Livesley, Robert Kenneth. An introduction to automatic digital computers. Cambridge University Press, 2017.
3. Zawacki-Richter, Olaf, and Colin Latchem. "Exploring four decades of research in Computers & Education." Computers & Education 122 (2018): 136-152.
4. Sinha, Pradeep K., and Priti Sinha. Computer fundamentals. BPB publications, 2010.
5. Goel, Anita. Computer fundamentals. Pearson Education India, 2010.

URCQ-5102 Discrete Structures

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	-

Course Introduction:

Introduces the foundations of discrete mathematics as they apply to Computer Science, focusing on providing a solid theoretical foundation for further work. Further, this course aims to develop understanding and appreciation of the finite nature inherent in most Computer Science problems and structures through study of combinatorial reasoning, abstract algebra, iterative procedures, predicate calculus, tree and graph structures. In this course more emphasis shall be given to statistical and probabilistic formulation with respect to computing aspects.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Understand the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs and Trees etc.	C2 (understand)
CLO-2	Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.	C3 (Apply)
CLO-3	Apply discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.	C3 (Apply)
CLO-4	Differentiate various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular	C4 (Differentiate)

Course Outline:

Mathematical reasoning, propositional and predicate logic, rules of inference, proof by induction, proof by contraposition, proof by contradiction, proof by implication, set theory, relations, equivalence relations and partitions, partial orderings, recurrence relations, functions, mappings, function composition, inverse functions, recursive functions, Number Theory, sequences, series, counting, inclusion and exclusion principle, pigeonhole principle, permutations and combinations. Algorithms, Searching and Sorting Algorithms, elements of graph theory, planar graphs, graph coloring, Graph Algorithms, euler graph, Hamiltonian path, rooted trees, traversals.

Reference Materials (or use any other standard and latest books):

1. Discrete Mathematics and Its Applications, 7th edition by Kenneth H. Rosen
2. Discrete Mathematics with Applications, 4th Edition by Susanna S. Epp
3. Discrete Mathematics, 7th edition by Richard Johnson Baugh
4. Discrete Mathematical Structures, 4th edition by Kolman, Busby & Ross
5. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi
6. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred Grassman

URCQ-5103 Calculus and Analytic Geometry

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	-

Course Introduction:

To provide foundation and basic ground for calculus and analytical geometry background.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
-	-	

Course Outline:

Limits and Continuity; Introduction to functions, Introduction to limits, Techniques of funding limits, Indeterminate forms of limits, Continuous and discontinuous functions and their applications, Differential calculus; Concept and idea of differentiation, Geometrical and Physical meaning of derivatives, Rules of differentiation, Techniques of differentiation, Rates of change, Tangents and Normals lines, Chain rule, implicit differentiation, linear approximation, Applications of differentiation; Extreme value functions, Mean value theorems, Maxima and Minima of a function for single-variable, Concavity, Integral calculus; Concept and idea of Integration, Indefinite Integrals, Techniques of integration, Riemann sums and Definite Integrals, Applications of definite integrals, Improper integral, Applications of Integration; Area under the curve, Analytical Geometry; Straight lines in R3, Equations for planes.

Reference Materials (or use any other standard and latest books):

1. Calculus and Analytic Geometry by Kenneth W. Thomas.
2. Calculus by Stewart, James.
3. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole

URCE-5118 Functional English

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	None

Course Introduction:

Course Introduction:

This is first course in English to the Bachelor of Science students and covers all the fundamental concept of English composition and comprehension. The course is designed in such a way that students can use this knowledge to further enhance their language skills in English. The course aims at enhancing students' skill and competence in communicating their ideas in writing and speaking in English language. It will primarily focus on four areas of language to help the students achieve proficiency in language use, develop skills in listening comprehension, improve reading efficiency, use the conventions of standard written English with skill and assertion, build-up vocabulary, and clearly and accurately reproduce specific data. It will illustrate the force and effectiveness of simple and direct English.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	-	-

Course Outline:

Paragraph and Essay Writing, Descriptive Essays; Sentence Errors, Persuasive Writing; How to give presentations, Sentence Errors; Oral Presentations, Comparison and Contrast Essays, Dialogue Writing, Short Story Writing, Review Writing, Narrative Essays, Letter Writing

Reference Materials (or use any other standard and latest books):

1. College Writing Skills with Readings, by John Langan, McGraw-Hill, 5th Edition.
2. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

URCS-5210 Digital Skills

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

This course introduces Digital skills which is basically the use of websites, apps, mobile devices, social media and its diverse platforms, search engines, and other digital means to promote and sell products and services

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand of digital skills, its principles, and best practices.	C2(Understand)
CLO-2	Apply theoretical concepts to real-world scenarios, demonstrating proficiency in executing effective digital skills and strategies.	C3(Apply)
CLO-3	Develop analytical skills to measure and optimize the performance of digital skill based initiatives, utilizing relevant metrics.	C4(Analyze)

Course Outline:

1- Introduction to fundamental concepts of digital skills, digital strategy and planning, emphasizing the use of websites, apps, mobile devices, social media and its diverse platforms, search engines, and other digital means. Ethical use of social media. Search engine optimization (SEO), social media marketing, email campaigns, and pay-per-click (PPC) advertising. Analytics and measurement techniques, legal and ethical considerations, and exploration of emerging trends. The use of latest online tools for better learning as IT student.

Reference Materials (or use any other standard and latest books):

1. "Basic Digital Skills: Your Guide to Basic Digital Skills" . A Training Manual by British Council. Second Edition, 2022.
2. "Digital Skills: Unlocking the Information Society" by Kennith A Leparo, Publisher: Palgrave Macmillan 2014.

CMPC-5202 Object Oriented Programming

Credit Hours:	4 (3-3)
Contact Hours:	3-3
Pre-requisites:	Programming Fundamentals

Course Introduction:

The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand principles of object-oriented paradigm.	C2 (Understand)
CLO-2	Identify the objects & their relationships to build object-oriented solution	C3 (Apply)
CLO-3	Model a solution for a given problem using object-oriented principles	C3 (Solve)
CLO-4	Examine an object-oriented solution	C4 (Examine)

Course Outline:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Reference Materials (or use any other standard and latest books):

1. Java: How to Program, 9th Edition by Paul Deitel
2. Beginning Java 2, 7th Edition by Ivor Horton

3. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu
4. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis
5. C++ How to Program, 10th Edition, Deitel & Deitel.
6. Object Oriented Programming in C++, 3rd Edition by Robert Lafore

CMPC-5204 Digital Logic Design

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

The course introduces the concept of digital logic, gates and the digital circuits. Further, it focuses on the design and analysis combinational and sequential circuits. It also serves to familiarize the student with the logic design of basic computer hardware components.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits	-
CLO-2	Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques	-
CLO-3	Apply the acquired knowledge to simulate and implement small-scale digital circuits	-
CLO-4	Understand the relationship between abstract logic characterizations and practical electrical implementations.	-

Course Outline:

Number Systems, Logic Gates, Boolean Algebra, Combination logic circuits and designs, Simplification Methods (K-Map, Quinn Mc-Cluskey method), Flip Flops and Latches, Asynchronous and Synchronous circuits, Counters, Shift Registers, Counters, Triggered devices & its types. Mealy machines and Moore machines. Binary Arithmetic and Arithmetic Circuits, Memory Elements, State Machines. Introduction Programmable Logic Devices (CPLD, FPGA) Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim.

Reference Materials (or use any other standard and latest books):

1. Digital Fundamentals by Floyd, 11/e.
2. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e

CMPC-5202 Object Oriented Programming

Credit Hours:	4 (3-3)
Contact Hours:	3-3
Pre-requisites:	Programming Fundamentals

Course Introduction:

The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand principles of object-oriented paradigm.	C2 (Understand)
CLO-2	Identify the objects & their relationships to build object-oriented solution	C3 (Apply)
CLO-3	Model a solution for a given problem using object-oriented principles	C3 (Solve)
CLO-4	Examine an object-oriented solution	C4 (Examine)

Course Outline:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Reference Materials (or use any other standard and latest books):

1. Java: How to Program, 9th Edition by Paul Deitel
2. Beginning Java 2, 7th Edition by Ivor Horton
3. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu
4. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis
5. C++ How to Program, 10th Edition, Deitel & Deitel.
6. Object Oriented Programming in C++, 3rd Edition by Robert Lafore

CMPC-5203 Database Systems

Credit Hours:	4 (3-3)
Contact Hours:	3-3
Pre-requisites:	None

Course Introduction:

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.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Explain fundamental database concepts.	C2 (Explain)

CLO-2	Design conceptual, logical and physical database schemas using different data models.	C5 (Design)
CLO-3	Identify functional dependencies and resolve database anomalies by normalizing database tables.	C2 (Identify)
CLO-4	Use Structured Query Language (SQL) for database definition and manipulation in any DBMS	C4 (Use)

Course Outline:

Basic database concepts, Database approach vs. file based system, database architecture, three level schema architecture, data independence, relational data model, attributes, schemas, tuples, domains, relation instances, keys of relations, integrity constraints, relational algebra, selection, projection, Cartesian product, types of joins, normalization, functional dependencies, normal forms, entity relationship model, entity sets, attributes, relationship, entity-relationship diagrams, Structured Query Language (SQL), Joins and subqueries in SQL, Grouping and aggregation in SQL, concurrency control, database backup and recovery, indexes, NoSQL systems.

Reference Materials (or use any other standard and latest books):

1. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition by Thomas Connolly and Carolyn Begg
2. Database Systems: The Complete Book, 2nd Edition by Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
3. Database System Concepts, 6th Edition by Avi Silberschatz, Henry F. Korth and S. Sudarshan.
4. Database Management Systems, 3rd Edition by Raghu Ramakrishnan, Johannes Gehrke

CMPC-5204 Digital Logic Design

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

The course introduces the concept of digital logic, gates and the digital circuits. Further, it focuses on the design and analysis combinational and sequential circuits. It also serves to familiarize the student with the logic design of basic computer hardware components.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits	-
CLO-2	Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques	-
CLO-3	Apply the acquired knowledge to simulate and implement small-scale digital circuits	-
CLO-4	Understand the relationship between abstract logic	-

	characterizations and practical electrical implementations.	

Course Outline:

Number Systems, Logic Gates, Boolean Algebra, Combination logic circuits and designs, Simplification Methods (K-Map, Quinn Mc-Cluskey method), Flip Flops and Latches, Asynchronous and Synchronous circuits, Counters, Shift Registers, Counters, Triggered devices & its types. Mealy machines and Moore machines. Binary Arithmetic and Arithmetic Circuits, Memory Elements, State Machines. Introduction Programmable Logic Devices (CPLD, FPGA) Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim.

Reference Materials (or use any other standard and latest books):

1. Digital Fundamentals by Floyd, 11/e.
2. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e

MATH-5101 Multivariable Calculus

Credit Hours:	3(3-0)	**Prerequisites:**	URCM-5108

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Develop the skills to have ground knowledge of multivariate calculus and appreciation for their further computer science courses.	MATH & SC	1

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

1. Multivariable Functions and Partial Derivatives: Functions of Several Variables. Limits and Continuity. Partial Derivatives. Differentiability, Linearization, and Differentials. The Chain Rule. Partial Derivatives with Constrained Variables. Directional Derivatives, Gradient Vectors, and Tangent Planes. Extreme Values and Saddle Points. Lagrange Multipliers. Taylor's Formula. [TB1: Ch. 11]
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. [TB1: Ch. 12]
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. Dirac's Delta Function. Partial Fractions. Convolution. Integral Equations. Differentiation and Integration of Transform. Systems of ODEs. Laplace Transform: General Formulas. Table of Laplace Transforms. [TB2: Ch. 6]
4. Fourier analysis: Fourier series, Arbitrary Period. Even and Odd Function. Half-Rang Expansions. Forced Oscillations. Approximation by Trigonometric Polynomials. StumLiouville Problems. Orthogonal Functions. Orthogonal Series. Generalized Fourier series. Fourier Integral. Fourier Cosine and Sine Transforms. Fourier Transform. [TB2: Ch. 11]

5. Power Series, Taylor Series: Sequences, Series, Convergence Tests. Power Series. Functions Given by Power Series. Taylor and Maclaurin Series. [TB2: Ch. 15]\
6. Laurent Series. Residue Integration: Laurent Series. Singularities and Zeros. Infinity. Residue Integration Method. Residue Integration of Real Integrals. [TB2: Ch. 16]

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Lab Assignments, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Calculus & Analytic Geometry by Thomas, Wiley; 10th Edition (August 16, 2011). ISBN-10: 0470458364
2. Advanced Engineering Mathematics by Erwin Kreyszig, Wiley; 10th Edition (August 16, 2011). ISBN-10: 0470458364
3. Multivariable Calculus by James Stewart, Brooks Cole; 7th Edition (January 1, 2011). ISBN-10: 0538497874
4. Multivariable Calculus by James Stewart 6th Edition, 2007, Cengage Learning publishers.
5. Calculus and Analytical Geometry by Swokowski, Olinick and Pence, 6th Edition, 1994, Thomson Learning EMEA, Ltd.
6. Elementary Multivariable Calculus by Bernard Kolman William F. Trench, 1971, Academic Press.
7. Multivariable Calculus by Howard Anton, Albert Herr 5th Edition, 1995, John Wiley

MATH-5102 Linear Algebra

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	Calculus and Analytical Geometry

Course Introduction:

To provide fundamentals of solution for system of linear equations, operations on system of equations, matrix properties, solutions and study of their properties.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Develops students fundamental skills of solving ordinary differential equations, and developing differential equations for real-world problems	C2(Understand)

Course Outline:

Algebra of linear transformations and matrices. determinants, rank, systems of equations, vector spaces, orthogonal transformations, linear dependence, linear Independence and bases, eigenvalues and eigenvectors, characteristic equations, Inner product space and quadratic forms.

Reference Materials (or use any other standard and latest books):

1. Introduction to Linear Algebra by Gilbert Strang, Wellesley Cambridge Press; latest Edition
2. Elementary Linear Algebra with Applications by Bernard Kolman, David Hill, latest Edition, Prentice Hall.

CMPC-5205 Data Structures

Credit Hours:	4 (3-3)
Contact Hours:	3-3
Pre-requisites:	Programming Fundamentals

Course Introduction:

The course is designed to teach students structures and schemes, which allow them to write programmer to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of time and space complexity of computer programs.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Implement various data structures and their algorithms and apply them in implementing simple applications	C3 (Apply)
CLO-2	Analyze simple algorithms and determine their complexities.	C5 (Analyze)
CLO-3	Apply the knowledge of data structure to other application domains.	C3 (Apply)
CLO-4	Design new data structures and algorithms to solve problems.	C6 (Design)

Course Outline:

Abstract data types, complexity analysis, Big Oh notation, Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms, Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix, bucket), queue, dequeuer, priority queues (linked and array implementations of queues), linked list & its various types, sorted linked list, searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining, trees and tree traversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first and depth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection.

Reference Materials (or use any other standard and latest books):

1. Data Structures and Algorithm Analysis in Java by Mark A. Weiss
2. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry
3. Data Structures and Algorithms in C++ by Adam Drozdek
4. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss Java Software Structures: Designing and Using Data Structures by John Lewis and Joseph Chase

CMPC-5206 Information Security

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

This course provides a broad overview of the threats to the security of information systems, the responsibilities and basic tools for information security, and the levels of training and expertise needed in organizations to reach and maintain a state of acceptable security. It covers concepts and applications of system and data security. Areas of particular focus include secure network design, implementation and transition issues, and techniques for responding to security breaches.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Explain key concepts of information security such as design principles, cryptography, risk management, and ethics	C2 (Explain)
CLO-2	Discuss legal, ethical, and professional issues in information security	A2 (Discuss)
CLO-3	Apply various security and risk management tools for achieving information security and privacy	C3 (Apply)
CLO-4	Identify appropriate techniques to tackle and solve problems in the discipline of information security	C4 (Identify)

Course Outline:

Information security foundations, security design principles; security mechanisms, symmetric and asymmetric cryptography, encryption, hash functions, digital signatures, key management, authentication and access control; software security, vulnerabilities and protections, malware, database security; network security, firewalls, intrusion detection; security policies, policy formation and enforcement, risk assessment, cybercrime, law and ethics in information security, privacy and anonymity of data.

Reference Materials (or use any other standard and latest books):

1. Computer Security: Principles and Practice, 3rd edition by William Stallings
2. Principles of Information Security, 6th edition by M. Whitman and H. Mattord
3. Computer Security, 3rd edition by Dieter Gollmann
4. Computer Security Fundamentals, 3rd edition by William Easttom
5. Official (ISC)2 Guide to the CISSP CBK, 3rd edition

CMPC-5207 Artificial Intelligence

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

Artificial Intelligence has emerged as one of the most significant and promising areas of computing. This course focuses on the foundations of AI and its basic techniques like Symbolic manipulations, Pattern Matching, Knowledge Representation, Decision Making and Appreciating the differences between Knowledge, Data and Code. AI programming language Python has been proposed for the practical work of this course.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand the fundamental constructs of Python programming language.	C2 (Understand)
CLO-2	Understand key concepts in the field of artificial intelligence	C2 (Understand)
CLO-3	Implement artificial intelligence techniques and case studies	C3 (Apply)

Course Outline:

An Introduction to Artificial Intelligence and its applications towards Knowledge Based Systems; Introduction to Reasoning and Knowledge Representation, Problem Solving by Searching (Informed searching, Uninformed searching, Heuristics, Local searching, Minmax algorithm, Alpha beta pruning, Game-playing); Case Studies: General Problem Solver, Eliza, Student, Macsyma; Learning from examples; ANN and Natural Language Processing; Recent trends in AI and applications of AI algorithms. Python programming language will be used to explore and illustrate various issues and techniques in Artificial Intelligence.

Reference Materials (or use any other standard and latest books):

1. Russell, S. and Norvig, P. "Artificial Intelligence. A Modern Approach", 3rd ed, Prentice Hall, Inc., 2015.
2. Norvig, P., "Paradigms of Artificial Intelligence Programming: Case studies in Common Lisp", Morgan Kaufman Publishers, Inc., 1992.
3. Luger, G.F. and Stubblefield, W.A., "AI algorithms, data structures, and idioms in Prolog, Lisp, and Java", Pearson Addison-Wesley. 2009.
4. Severance, C.R., 2016. "Python for everybody: Exploring data using Python 3." CreateSpace Independent Publ Platform.
5. Miller, B.N., Ranum, D.L. and Anderson, J., 2019. "Python programming in context." Jones & Bartlett Pub.
6. Joshi, P., 2017. "Artificial intelligence with python." Packt Publishing Ltd.

CMPC-5208 Computer Networks

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

This course introduces the basic concept of computer network to the students. Network layers, Network models (OSI, TCP/IP) and protocol standards are part of the course.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Describe the key terminologies and technologies of computer	C2 (Describe)

	networks	
CLO-2	Explain the services and functions provided by each layer in the Internet protocol stack.	C2 (Explain)
CLO-3	Identify various internetworking devices and protocols and their functions in a networking	C4 (Identify)
CLO-4	Analyze working and performance of key technologies, algorithms and protocols	C4 (Analyze)
CLO-5	Build Computer Network on various Topologies	P3 (Build)

Course Outline:

Introduction and protocols architecture, basic concepts of networking, network topologies, layered architecture, physical layer functionality, data link layer functionality, multiple access techniques, circuit switching and packet switching, LAN technologies, wireless networks, MAC addressing, networking devices, network layer protocols, IPv4 and IPv6, IP addressing, sub netting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks.

Reference Materials (or use any other standard and latest books):

1. Computer Networking: A Top-Down Approach Featuring the Internet, 6th edition by James F. Kurose and Keith W. Ross
2. Computer Networks, 5th Edition by Andrew S. Tanenbaum
3. Data and Computer Communications, 10th Edition by William Stallings
4. Data Communication and Computer Networks, 5th Edition by Behrouz A. Forouzan

CMPC-5101 Software Engineering

Credit Hours:	3 (3-0)
Contact Hours:	3-0
Pre-requisites:	None

Course Introduction:

This course provides students with a foundational understanding of the principles, methodologies, and practices essential for designing, developing, and maintaining software systems. Emphasis is placed on the entire software development life cycle, covering requirements analysis, system design, quality assurance and testing.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Describe various software engineering processes and activates	C1 (Describe)
CLO-2	Apply the system modeling techniques to model a medium size software systems	C3 (Apply)
CLO-3	Apply software quality assurance and testing principles to medium size software systems	C4 (Apply)
CLO-4	Discuss key principles and common methods for software project management such as scheduling, size estimation, cost	C2 (Discuss)

	estimation and risk analysis	

Course Outline:

Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software process models, Agile software Development, Agile process models, Agile development techniques, Requirements engineering process, Functional and non-functional requirements, Context models, Interaction models, Structural models, behavioral models, model driven engineering, Architectural design, Design and implementation, UML diagrams, Design patterns, Software testing and quality assurance, Software evolution, Project management and project planning, configuration management, Software Process improvement

Reference Materials (or use any other standard and latest books):

1. Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014
2. Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.

MATH-5103 Probability and Statistics

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	None

Course Introduction:

To introduce the concepts of data analysis, presentation, counting techniques, probability and decision making.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Develops students fundamental skills of solving ordinary differential equations, and developing differential equations for real-world problems	C2(Understand)

Course Outline:

Introduction to Statistics and Data Analysis, Statistical Inference, Samples, Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule. Random Variables and Probability Distributions. Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem. Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of S2, t-Distribution, FQuantile and Probability Plots. Single Sample & One- and Two-Sample Estimation Problems. Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of PValues for Decision Making in Testing Hypotheses (Single Sample & One- and TwoSample Tests), Linear

Regression and Correlation. Least Squares and the Fitted Model, Multiple Linear Regression and Certain, Nonlinear Regression Models, Linear Regression Model Using Matrices, Properties of the Least Squares Estimators.

Reference Materials (or use any other standard and latest books):

1. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116
2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, Duxbury Press; 3rd Edition (February 3, 2006), ISBN-10:0495107573
3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259

CMPC-5209 Computer Organization & Assembly Language

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	Digital Logic Design

Course Introduction:

The main objective of this course is to introduce the organization of computer systems and usage of assembly language for optimization and control. Emphasis should be given to expose the low-level logic employed for problem solving while using assembly language as a tool. At the end of the course the students should be capable of writing moderately complex assembly language subroutines and interfacing them to any high level language.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Acquire the basic knowledge of computer organization computer architecture and assembly language	C2 (Understand)
CLO-2	Understand the concepts of basic computer organization, architecture, and assembly language techniques	C2 (Understand)
CLO-3	Solve the problems related to computer organization and assembly language	C3 (Apply)

Course Outline:

Introduction to computer systems: Information is bits + context, programs are translated by other programs into different forms, it pays to understand how compilation systems work, processors read and interpret instructions stored in memory, caches matter, storage devices form a hierarchy, the operating system manages the hardware, systems communicate with other systems using networks; Representing and manipulating information: information storage, integer representations, integer arithmetic, floating point; Machine-level representation of programs: a historical perspective, program encodings, data formats, accessing information, arithmetic and logical operations, control, procedures, array allocation and access, heterogeneous data structures, putting it together: understanding pointers, life in the real world: using the gdb debugger, out of-bounds memory references and buffer overflow, x86-64: extending ia32 to 64 bits, machine-level representations of floating-point programs; Processor architecture: the Y86 instruction set architecture, logic design and the Hardware Control Language (HCL), sequential Y86 implementations, general principles of pipelining, pipelined Y86 implementations

Reference Materials (or use any other standard and latest books):

1. Computer System Architecture, M. Morris Mano, Latest Edition,
2. Assembly Language Programming for Intel- Computer, Latest Edition
3. Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e), Randal E. Bryant and
4. David R.O' Hallaron, Carnegie Mellon University
5. Robert Britton, MIPS Assembly Language Programming, Latest Edition

SEDC-5101 Software Design & Architecture

Credit Hours:	3(3-0)	**Prerequisites:**	-

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process.	C	2
2. Comprehend the advantages of consistent and reliable software design.	C	3
3. Design OOD models and refine them to reflect implementation details	C	3
4. Apply and use UML to visualize and document the design of software systems.	C	3
5. Implement the design model using an object-oriented programming language.	C	5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

1. Software Architecture (SA): SA in Context, SA as a Design Plan, Abstraction, SA Terminology, 4+1 Views Model: Coupling Between Views, Uses and Notation of Four Views. IS2000: The Advanced Imaging Solution, Chapter 1 [TB 1:Ch. 1, Ch. 2]
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 [TB 2: Ch.1, Ch.2]
3. SA Case Study: A-7E Avionics System - Architecture, Business Cycle, Requirements and Qualities [TB 2: Ch. 3]
4. Creating Architecture: Understanding Quality Attributes, Functionality and Architecture, Architecture and Quality Attributes, System Quality Attributes, Quality Attribute Scenarios in Practice, Other System Quality Attributes, Business Qualities, Architecture Qualities [TB 2: Ch. 4]
5. Achieving Qualities: Introducing Tactics, Availability Tactics, Modifiability Tactics, Performance Tactics, Security Tactics, Testability Tactics, Usability Tactics, Relationship of

Tactics to Architectural Patterns, Architectural Patterns and Styles [TB 2: Ch. 5]

6. Global Analysis: Overview of Global Analysis Activities, Analyze Factors, Develop Strategies., Analyze Organizational Factors, Begin Developing Strategies, Analyze Technological Factors, Continue Developing Strategies, Analyze Product Factors, Continue Developing Strategies [TB 1: Ch. 3]
7. Conceptual Architecture View: Design Activities for the Conceptual Architecture View, Global Analysis, Central Design Tasks: Components, Connectors, and Configuration, Final Design Task: Resource Budgeting, Traceability, Uses for the Conceptual Architecture View [TB 1: Ch. 4]
8. Module Architecture View: Design Activities for the Module Architecture View, Global Analysis, Central Design Tasks: Modularization and Layering, Final Design Task: Interface Design, Traceability, Uses for the Module Architecture View [TB 1: Ch. 5]
9. Execution Architecture View: Design Activities for the Execution Architecture View, Global Analysis, Central Design Tasks: Runtime Entities, Communication Paths, and Configuration, Final Design Task: Resource Allocation, Traceability, Uses for the Execution Architecture View [TB 1: Ch. 6]
10. Code Architecture View: Design Activities for the Code Architecture View, Global Analysis, Central Design Tasks, Final Design Tasks, Traceability, Uses for the Code Architecture View [TB 1: Ch. 7]
11. Designing & Documenting the Architecture: Architecture in the Life Cycle, Designing the Architecture, Forming Team, Creating a Skeletal System. Documenting Software Architectures, Uses of Architectural Documentation, Views, Choosing the Relevant Views, Documenting a View, Documentation across Views, Unified Modelling Language [TB 2: Ch. 7, Ch. 9]
12. 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 [TB 2: Ch. 11]
13. The CBAM: A Quantitative Approach to Architecture Design Decision Making, Decision-Making Context, The Basis for the CBAM, Implementing the CBAM, Case Study: The NASA ECS Project, Results of the CBAM Exercise [TB 2: Ch. 12]

    Reconstructing Software Architectures: Introduction, Information Extraction, Database Construction, View Fusion, Reconstruction [TB 2: Ch. 10]

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Applied Software Architecture by Christine Hofmeister, Robert Nord and DilipSoni, Addison-Wesley Professional (1999). ISBN-10: 0201325713.
2. Software Architecture in Practice by Len Bass, Paul Clements and Rick Kazman, Addison-Wesley Professional; 2ndEdition (April 19, 2003). ISBN-10: 0321154959
3. Software Architecture in Practice by Len Bass, Paul Clements and Rick Kazman, Addison-Wesley Professional; 3rdEdition (2012). ISBN-10: 0321815734

4. Software Architecture and Design Illuminated by Kai Qian, Xiang Fu, Lixin Tao and Chong-weiXu, Jones & Bartlett Publishers; 1stEdition (2009). ISBN-10: 076375420X
5. Software Architecture: Foundations, Theory, and Practice by R. N. Taylor, N. Medvidovic and E. M. Dashofy, Wiley; 1stEdition (2009). ISBN-10: 0470167742.

SEDC-5201 Software Construction & Development

Credit Hours:	3(2-3)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. To apply a wide variety of software construction techniques and tools, including state-based and table-driven approaches to low-level design of software.	C	2
2. Able to state-based and table-driven approaches to low-level design of software.	C	2
3. To understand and apply collaborative construction	C	2
4. To understand Refactoring & its strategies	C	2
5. To understand layout and styling of developing software	C	2

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

1. Software Construction: What Is Software Construction? Why Is Software Construction Important? Metaphors for Software Development, The Importance of Metaphors, how to Use Software Metaphors, Common Software Metaphors. [TB1: Ch. 1, 2]
2. Prerequisites: Importance of Prerequisites, Type of Target Software, Problem-Definition Prerequisite, Requirements Prerequisite, Architecture Prerequisite, Time Constraints. [TB1: Ch. 3]
3. Key Construction Decisions: Choice of Programming Language, Programming Conventions, Localization Aspects of Technology, Selection of Construction Practices. [TB1: Ch. 4]
4. Design in Software Construction: Design Challenges, Key Design Concepts, Design Building Blocks: Heuristics, Design Practices, Popular Methodologies. [TB1: Ch. 5]
5. Defensive Programming: Protecting Your Program from Invalid Inputs, Assertions, Error Handling Techniques, Exceptions, Barricade Your Program to Contain the Damage Caused by Errors, Debugging Aids, Determining How Much Defensive Programming to Leave in Production Code, Being Defensive About Defensive Programming, The Pseudocode. [TB1: Ch. 8]
6. The Software-Quality Landscape: Characteristics of Software Quality, Techniques for Improving Software Quality, Relative Effectiveness of Quality Techniques, when to Do Quality Assurance, Principle of Software Quality. [TB1: Ch. 20]
7. Collaborative Construction: Overview of Collaborative Development Practices, Pair Programming, Formal Inspections, Other Kinds of Collaborative Development Practices. [TB1: Ch. 21]
8. Refactoring: Kinds of Software Evolution, Introduction to Refactoring, Reasons to Refactor,

Specific Refactoring, Refactoring Safely, Refactoring Strategies. [TB1: Ch. 24]

9. Program Size & Software Construction: Communication and Size, Range of Project Sizes, Effect of Project Size on Errors, Effect of Project Size on Productivity, Effect of Project Size on Development Activities. [TB1: Ch. 27]
10. Managing Construction: Encouraging Good Coding, Configuration Management, estimating a Construction Schedule, Measurement, Treating Programmers as People, 8.6 Managing Your Manager. [TB1: Ch. 28]
11. Integration: Importance of the Integration Approach, Integration Frequency—Phased or Incremental? Incremental Integration Strategies, Daily Build and Smoke Test. [TB1: Ch. 29]
12. Programming Tools: Design Tools: Source-Code Tools, Executable-Code Tools, Tool-Oriented Environments, Building Your Own Programming Tools, Tool Fantasyland. [TB1: Ch. 30]
13. Layout and Style: Layout Fundamentals, Layout Techniques, Layout Styles, Laying Out Control Structures, Laying Out Individual Statements, Laying Out Comments, Laying Out Routines, Laying Out Classes. [TB1: Ch. 31]
14. Self-Documenting Code: External Documentation, Programming Style as Documentation, To Comment or Not to Comment, Keys to Effective Comments, Commenting Techniques. [TB1: Ch. 32]

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Code Complete: A Practical Handbook of Software Construction by Steve McConnell, Microsoft Press; 2nd Edition (July 7, 2004). ISBN-10: 0735619670
2. Compiler Construction (International Computer Science Series) by Niklaus Wirth, Addison-Wesley Pub (Sd); (1996). ISBN-10: 0201403536.
3. Object-Oriented Software Construction (Book/CD-ROM) (2nd Edition) by Bertrand Meyer, Prentice Hall; 2nd Edition (2000). ISBN-10: 0136291554.
4. The Design of Well-Structured and Correct Programs, S. Alagic& M.A. Arbib, Springer-Verlag (1978), ISBN 0-387-90299-6.
5. Object-Oriented Software Construction, by Bertrand Meyer, Second Edition, Published by, Prentice Hall in 1997. Prentice Hall; 2nd Edition (March 21, 2000). ISBN-10: 0136291554

URCW-5201 Applied Physics

Credit Hours:	3 (2-3)
Contact Hours:	2-3
Pre-requisites:	None

Course Introduction:

The course introduces students with the basic concept of Physics and electronics. Students are also taught Physics laws and other associate topics to prepare them for the advanced level courses in this area. The focus of the course on electric force and its applications and related problems, conservation of charge, charge quantization, Electric fields due to point charge and lines of force and many other useful topics.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
-	-	

Course Outline:

Electric force and its applications and related problems, conservation of charge, charge quantization, Electric fields due to point charge and lines of force. Ring of charge, Disk of charge, A point charge in an electric field, Dipole in a n electric field, The flux of vector field, The flux of electric field, Gauss' Law, Application of Gauss' Law, Spherically symmetric charge distribution, A charge isolated conductor, Electric potential energy, Electric potentials, Calculating the potential from the field and related problem Potential due to point and continuous charge distribution, Potential due to dipole, equipotential surfaces, Calculating the field from the potential, Electric current, Current density, Resistance, Resistivity and conductivity, Ohm's law and its applications, The Hall effect, The magnetic force on a current, The Biot-Savart law, Line of B, Two parallel conductors, Amperes' s Law, Solenoid, Toroids, Faraday's experiments, Faraday's Law of Induction, Lenz's law, Motional emf, Induced electric field, Induced electric fields, The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves, Total internal reflection, Two source interference, Double Slit interference, related problems, Interference from thin films, Diffraction and the wave theory, related problems, Single-Slit Diffraction, related problems, Polarization of electromagnetic waves, Polarizing sheets, related problems.

Reference Materials (or use any other standard and latest books):

1. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker
2. Narciso Garcia, Arthur Damask, Steven Schwarz., "Physics for Computer Science Students", Springer Verlag, 1998.

URCE-5119 Expository Writing

Credit Hours:	3 (3-0)
Contact Hours:	3
Pre-requisites:	Functional English

Course Introduction:

The course introduces students to the communications so they can effectively communicate their message. The course also covers how to make an effective presentation both written and verbal. Various modern techniques of communication and presentation skills are covered in this course. Further the course aims to enhance students' linguistic command, so they could communicate effectively in diversified socio-cultural situations; create larger stretches of interactive text in speech and writing; and identify and repair any instances of potential communication break-up.

CLO	Course Learning Outcomes	Bloom

No.		Taxonomy
CLO-1	-	-

Course Outline:

Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams; Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation.

Reference Materials (or use any other standard and latest books):

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748
3. College Writing Skills with Readings, by John Langan, McGraw-Hill, 5th Edition.
4. A Textbook Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

URCI-5105 Islamic Studies

Credit Hours: **Contact Hours:** **Pre-requisites:**	2 (2-0) 2 -

Course Introduction:

To provide Basic information about Islamic Studies. To enhance understanding of the students regarding Islamic Civilization. History of Islam, understanding of the worship and its usefulness. The basic concept of Quran Pak: wisdom, patience, loyalty. The comparative analysis of Islam with other religions. The Concept and Value of ***Haqooq ul Ibad*** (Bandon Kay Haqooq) in Islam. What is The rights of people in Islamic Point of View. Islamic point of view about other religions.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	To further enhance the knowledge of Islam.	
CLO-2	To understand the basic concept of Islam and Quran Pak.	
CLO-3	To understand the concept of Haqooq ul ibad in the light of Quran.	
CLO-4	To know the importance of Islamic concept about other religions.	

Course Outline:

Basic Themes of Quran, Introduction to Sciences of Hadith, Introduction to Islamic Jurisprudence, Primary & Secondary Sources of Islamic Law, Makken & Madnian life of the Prophet, Islamic Economic System, Political theories, Social System of Islam. Definition of Akhlaq.The Most Important Characters mentioned in the Holy Qur'an and Sunnah, SIDQ (Truthfulness)Generosity Tawakkaul(trust on Allah)Patience Taqua (piety). Haqooq ul ibad in the light of Quran & Hadith - the important characteristic of Islamic Society.

Reference Materials (or use any other standard and latest books):

1. Introduction to Islam by Dr Hamidullah, Papular Library Publishers Lahore
2. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI
3. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services

URCQ-5111: Translation of the Holy Quran - I

Credit Hours:	NC	**Prerequisites:**	None
Semester / Level	**Insome discipline1stsemesterandinsomediscipline2ndSemester/ ADP Program1stYear**		
Course Learning Outcomes (CLOs):	• **Tofamiliarizethestudentstokeysandfundamentalsofrecitationof the holyQuran.** • **Todeveloptheskillofthestudentsofrecitationthelastrevelation.** • **StudentswilllearnthebasicArabicgrammarinapracticalway.** • **TodevelopaneagernessamongthestudentstoexplorethelastdivineBook.**		
Cours eCont ents:	• تیسواں پارہ - ناظرہ مع تجوید • بنیادی عربی گرامر اسم اور اسکے متعلقات : اسم فاعل ،مفعول ،تفضیل ،مبالغہ فعل اور اسکی اقسام : ماضی ،مضارع ،امر ، نہی حرف اور اسکی اقسام : حروف علت ،حروف جارہ ،مشبہ بالفعل		
Memorization:	تیسویں پارے کی آخری بیس سورتیں (حفظ مع ترجمہ)		

URCQ-5111: Translation of the Holy Quran - II

Credit Hours:	NC	**Prerequisites:**	None
Semester/Level	Insome discipline3rdsemesterandinsomediscipline4thSemester/ ADP Program2ndYear		

Course Learning Outcome (CLO):	▪ Students will come to know about the real nature, significance andrelevanceoftheIslamicbeliefsinlightofthetext oftheHolyQuran. ▪ Students will seek knowledge of translation and transliteration of theHolyBookQuran. ▪ TofamiliarizethestudentswiththeconceptofIbādah(Itssignificance ,scopeandrelevance)andits typesin Islam. ▪ Students will learn literal and idiomatic way of translation of the HolyBook. ▪ StudentswilllearnaboutthepolytheismanditsincompatibilityinIslamhighlightedbythe HolyQuran. ▪ Tohighlightthesignificanceoflearningthroughusingallhumanfaculties provided by the almighty Allah and familiarize the studentsaboutcondemnationofignorancementionedintheQuranictext. ▪ To develop Awareness among the students about rights and duties ofdifferentcirclesof societyin thelight of HolyQuran. ▪ TointroducethestudentstoQuranicArabicgrammar inpracticalmanner.
CourseContents:	oایمانیات اور عبادات بللا پر ایمان ،فرشتوں پر ایمان ،رسولوں پر ایمان ،آسمانی کتابوں پر ایمان یوم آخرت پر ایمان ، تقدیر پر ایمان نماز ،روزہ ،زکوۃ،حج ،جہاد • معاشرے کے حقوق • خاندان کی تکوین • حق مہر • رضاعت و حمل • اوالد کو قتل کرنے کے ممانعت • شوہرکی نافرمانی • طالق • بیوہ کی عدت کے احکام • نکاح کا پیغام بھیجنا • عورت کی وراثت)اس کے شوہر کی طرف سے(• والدین کے حقوق • بیویوں اور اوالد کے بیچ عداوت • خاندان کے حقوق • مہمان کی عزت • اجازت طلب کرنے کے اصول • مجلس کے آداب • تعاون اور بھائی چارہ • گروہ بندی • محبت • لوگوں کے درمیان صلح • عفو ودرگزر، غصہ پر قابو اور معاف کرنا • شعوب و قبائل • لوگوں کے بیچ اختالفات • حمایت و نگہبانی
Grammar:	• قرآنی عربی گرامر کے اصول اور انکے اطلاقات (متن قرآنی پر اطالق سے توضیحات)
Details ofChapters andverseNumbers:	• منتخب آیات مع ترجمہ وتجوید • □ البقرہ ((۱۱۷، ۲۳۸، ۴۵، ۱۱۸، ۲۷۸، ۱۷۷، ۴۵، ۳۴۷، □ ۱۵۸، ۷۱۸، ۴۲۸، ۵۳، ۴۲۸، • ۴۷، ۲۸۷، ۳۴۸، ۲۲، ۸۲، ۲۸۷، ۴۲۸، ۲۴۸، ۱۱۷، ۲۳۷،

۲۲۷، ۱۱۸، ۵۲۷، ۳۲۷، • ۲۷۸، ۲۴۷، ۲۲۷،۷۵۸، ۱۸۸، ۲۸۸، ۱۳۸، ۲۸۸، ۲۸۸، ۷۳۸، ۱۳۸، ۴۳۸، ۱۳۸ ، • ۸۲ ، ۵۲۷، ۳۳۸، ۸۲۷، ۱۲۷، ۳۲) • النساء (۴۲، ۸۲، ۲۴، ۲۳۷، ۲۲، ۱۲، ۳۷، ۲۲، ۱۲،۲۲، ۲۴، ۲۱۷، ۸۷، ۴۳، ۲۸۷، • ۵۳، ۷۷، ۵، ۲۱۷، ۵۸، ۴۸، ۷۷، ۷۷، ۲۸۷، ۷۲۷، ۴۳، ۱۸، ۴۳، ۷۲۷، ۲، ۷، ۱۸، ۷، • ۲۷، ۲۵۷، ۵۴، ۲۸۷، ۷۲) • االنعام (۸۸، ۸۲،۵۷،۱۳۷، ۷۴۷، ۲۵، ۵۴) • ال عمران (۱۲، ۲۳، ۴۲، ۴۸۷، ۵۲، ۵۵۷، ۲۷،۷) • المائده (۵۴، ۸، ۸۲، ۲۳،۲۷، ۸۲، ۸، ۴) • العراف (۴۳، ۲۲۷، ۲۲۷) • التوبه (۱۸، ۷۱،۲۷) • بود (۸۷) • الزمر (۲) • النور (۵۴، ۲۸،۸۴، ۱۸، ۱۲، ۲۸) • محمد (۳۳) • انفال (۸۲، ۱۸) • الرعد (۳) • الطالق (۵) • الحج(۴) • ابراہیم ((۳۸،۵۵) • السراء (۳۸، ۵۸) • الحقاف (۴۷) • المومنون (۱۸) • العنکبوت (۴۸،۲۵، ۲) • النحل (۸۸) • لقمان (۴۷،۵، ۵۷) • الحزاب ((۴۳، ۱۴،۲۳، ۲۵) • الشعراء (۱) • الروم (۷۸) • مریم (۵۷،۲۸) • المجادله (۸۷، ۷۷)	