

UNIVERSITY OF THE PUNJAB

NOTIFICATION

The Vice-Chancellor has, in exercise of the powers vested in him under Section 15(3) of the University of the Punjab Act, 1973 and in anticipation of approval of the Academic Council/Syndicate, been pleased to approve the recommendations of the Boards of Studies in Software Engineering, Information Technology and Computer Science and Board of Faculty of Computing & Information Technology at their meetings held on 10-04-2026, 23-04-2026, 30-04-2026 and 04-06-2026 respectively regarding the launch and approval of the Curriculum for the following Programs to be offered at University of the Punjab and its Affiliated Colleges from Fall 2026 :-

1. BS Computer Science
2. BS Computer Science (Specialization in Information Technology)
3. BS Computer Science (Specialization in Software Engineering)
4. BS Computer Science (Specialization in Artificial Intelligence)
5. Associate Degree in Computing
6. Associate Degree in Computing (Specialization in Information Technology)
7. Associate Degree in Computing (Specialization in Artificial Intelligence)

The Curriculum and Course Syllabi for above mentioned Programs are attached herewith, vide Annexure 'A' to 'G'.

**Admin. Block,
Quaid-i-Azam Campus,
Lahore.
No. D/ 2690 /Acad.**

Sd/-
**Dr. Ahmad Islam
Registrar**

Dated: 10-06-2026.

Copy of the above is forwarded to the following for information and further necessary action: -

1. Pro-Chancellor/The Minister of Education,
Govt. of the Punjab, Lahore.
2. Members of the Syndicate.
3. Dean, Faculty of Computing & Information Technology.
4. Chairman, Department of Computer Science/Software Engineering/
Information Technology
5. Principals of Affiliated Colleges (Concerned)
6. Controller of Examinations
7. Director. Quality Enhancement Cell
8. Director, IT (for Uploading on website)
9. Additional Registrar (Affiliation)
10. Deputy Registrar (General)
11. Secretary to the Vice-Chancellor
12. Secretary to the Pro-Vice-Chancellor
13. Secretary to the Registrar
14. Assistant Registrar (Statutes)
15. Assistant Registrar Syndicate (with file)
16. Assistant Registrar (Syllabus)



**Dr. Ahmad Islam
Registrar**

BS Computer Science (Specialization in Software Engineering)

Program Structure, Program and Course Learning Outcomes,
Continuous Quality Improvement (CQI) Process, Eligibility
Criteria, Scheme of Studies, and Course Outlines

**Department of Software Engineering
Faculty of Computing & Information Technology
University of the Punjab, Lahore**

Vision Statement [Department of Software Engineering]

To achieve excellence in software engineering education and research by producing competent graduates who contribute to technological advancement and national development.

Mission Statement [Department of Software Engineering]

The mission of the Department of Software Engineering at University of the Punjab is to provide quality education in software engineering through a balanced curriculum that integrates strong theoretical foundations with practical skills. The department strives to develop competent graduates with problem-solving abilities, professional ethics, and a commitment to lifelong learning. Through effective teaching, research, and industry engagement, the department aims to contribute to technological advancement and support national development.

Program Title

The title of the program is BS Computer Science (Specialization in Software Engineering).

The Program Introduction

The BS Computer Science (Specialization in Software Engineering) program is designed to develop graduates with strong theoretical foundations and practical skills in computing and software development. The program aims to equip students with knowledge of core computer science principles, including programming, data structures, computer architecture, databases, networks, artificial intelligence, and software engineering practices, while enabling them to specialize in the design, development, testing, and maintenance of large-scale software systems.

In alignment with the curriculum framework developed by the Higher Education Commission¹ of Pakistan, the program emphasizes problem-solving, algorithmic thinking, system design, and professional software engineering practices. The curriculum integrates computing fundamentals, mathematics, interdisciplinary courses, and domain electives to prepare graduates for professional careers in the rapidly evolving digital and software industry.

The specialization in Software Engineering provides focused knowledge in areas such as software design and architecture, requirements engineering, software project management, quality assurance, and modern development methodologies. Through laboratory work, industry-oriented projects, internships, and a capstone project, the program ensures that students gain practical experience and develop the competencies required to design reliable and scalable software solutions.

Vision Statement [BS Computer Science (Specialization in Software Engineering)]

To become a leading program in computer science education with specialization in software engineering by producing competent professionals who develop innovative, reliable, and socially responsible software solutions and contribute to the advancement of the computing profession.

¹ Higher Education Commission (HEC), Government of Pakistan. Curriculum of Computer Science. Subject: Revised Curriculum for BS Computer Science Programs. Diary No.: 8-36/HEC/Acad/2025 (Curriculum). Date: October 14, 2025

Mission Statement [BS Computer Science (Specialization in Software Engineering)]

To produce competent computing professionals with strong foundations in computer science and specialized expertise in software engineering, capable of designing and developing reliable software systems, solving complex real-world problems, working effectively in teams, and upholding professional, ethical, and societal responsibilities while pursuing lifelong learning.

Program Educational Objectives (PEOs)

PEO-1: Professional Practice

Graduates will apply computer science and software engineering principles to design, develop, and maintain quality software systems in professional computing environments.

PEO-2: Problem Solving and Innovation

Graduates will analyze complex computing problems and develop innovative software-based solutions using modern tools and technologies.

PEO-3: Professionalism and Teamwork

Graduates will demonstrate effective communication, teamwork, leadership, and adherence to professional and ethical standards in their careers.

PEO-4: Lifelong Learning and Adaptability

Graduates will engage in lifelong learning to adapt to emerging technologies, pursue advanced studies, and contribute to the evolving field of computing.

Program Learning Outcomes (PLOs)

BS Computer Science (Specialization in Software Engineering) prepare students to attain educational objectives by ensuring that students demonstrate achievement of the following outcomes (derived from Graduate Attributes define by Seoul Accord www.seoulaccord.org).

PLO-1: Academic Education

To prepare graduates as computing professionals.

PLO-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.

PLO-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.

PLO-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.

PLO-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.

PLO-6: Individual and Team Work

Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.

PLO-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.

PLO-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.

PLO-9: Ethics

Understand and commit to professional ethics, responsibilities, and norms of professional computing practice.

PLO-10: Life-long Learning

Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

Mapping of Program Learning Outcomes (PLOs) to Program Educational Objectives (PEOs)

This table illustrates the alignment between the Program Learning Outcomes (PLOs) and the Program Educational Objectives (PEOs) for the BS Computer Science (Software Engineering Specialization) program. Each PLO represents a specific competency that students are expected to achieve upon completion of the program, while each PEO reflects the broader career and professional goals that the program aims to fulfill. The mapping demonstrates how the program outcomes contribute to achieving the defined educational objectives, ensuring that graduates possess the knowledge, skills, and professional attributes necessary for academic, technical, and societal success.

PLOs	PEO-1	PEO-2	PEO-3	PEO-4
PLO-1 Academic Education	✓			
PLO-2 Knowledge for Solving Computing Problems	✓	✓		
PLO-3 Problem Analysis	✓	✓		
PLO-4 Design / Development of Solutions	✓	✓		
PLO-5 Modern Tool Usage	✓	✓		✓
PLO-6 Individual and Team Work			✓	
PLO-7 Communication			✓	
PLO-8 Computing Professionalism and Society			✓	
PLO-9 Ethics			✓	
PLO-10 Life-long Learning				✓

Continuous Quality Improvement (CQI) Procedure

Purpose

The purpose of this CQI procedure is to establish a systematic, data-driven, and sustainable mechanism to continuously monitor, evaluate, and enhance the quality of the BS Computer Science (Software Engineering specialization) program. It ensures alignment with Program Educational Objectives (PEOs), Program Learning Outcomes (PLOs), and evolving industry and academic requirements.

CQI Framework

The CQI process follows a closed-loop model: Plan → Deliver → Assess → Evaluate → Improve

Planning Phase

The planning phase involves defining and periodically reviewing the Program Educational Objectives (PEOs), Program Learning Outcomes (PLOs), and Course Learning Outcomes (CLOs) to ensure their continued relevance and alignment with academic and industry needs. It also includes establishing clear CLO–PLO mappings in accordance with Bloom’s Taxonomy to maintain appropriate cognitive levels. Furthermore, this phase encompasses the design and refinement of course content, assessment methods, and instructional strategies to support effective teaching and learning outcomes.

Delivery Phase

The delivery phase involves executing teaching activities in accordance with the approved course outlines while ensuring effective coverage of planned content. Faculty employ modern pedagogical approaches such as problem-based learning, project-based learning, and the use of industry-relevant case studies to enhance student engagement and practical understanding. Additionally, proper documentation is maintained through updated course files and teaching records to ensure transparency and support the overall quality assurance process.

Assessment Phase

The assessment phase comprises both direct and indirect evaluation methods to measure student learning and program effectiveness. Direct assessment includes midterm and final examinations, quizzes, assignments, lab work, projects, and capstone project evaluations to assess the attainment of course learning outcomes. Indirect assessment involves collecting feedback through course evaluation surveys, exit surveys, alumni surveys, and employer feedback to gain insights into student satisfaction and the overall impact of the program.

Evaluation Phase

The evaluation phase involves computing the attainment levels of Course Learning Outcomes (CLOs) at the course level and Program Learning Outcomes (PLOs) at the program level using collected assessment data. These results are then compared against predefined Key Performance Indicators (KPIs). Based on this analysis, weak CLOs and PLOs are identified, along with trends and recurring deficiencies, to inform subsequent improvement actions.

Improvement Phase

The improvement phase involves proposing corrective actions based on the findings of the evaluation process to address identified gaps and enhance program quality. These actions may include revising course content, modifying assessment

methods, organizing faculty training and workshops, and introducing new tools or technologies. The approved actions are implemented in the subsequent academic cycle, and all changes, along with their outcomes, are properly documented to ensure effective closing of the CQI loop.

CQI Cycle Frequency

The CQI cycle will be implemented at multiple levels to ensure continuous monitoring and improvement of the program. Course-level CQI will be conducted every semester to assess and enhance individual course outcomes, while program-level CQI will be carried out annually to evaluate overall program performance. Additionally, Program Educational Objectives (PEOs) will be reviewed every two to three years to ensure their continued relevance and alignment with evolving academic and industry requirements.

Eligibility Criteria

- Higher Secondary School Certificate/A-levels (involving minimum 12 years of schooling) or an IBCC equivalent qualification, with at least 50% marks and having studied Mathematics, is the basic eligibility requirement for admission in BS Computer Science (Specialization in Software Engineering).
 - The students who have not studied Mathematics at intermediate level have to pass deficiency courses of Mathematics (06 credits) in first year including zero semester.
- Additionally, applicants may be required to pass an entry test conducted by the department, university, or admission authority, along with fulfilling any other criteria set by the institution, such as interviews or aptitude assessments.

Program Structure

The program structure of BS Computer Science (Specialization in Software Engineering) is designed in accordance with the guidelines provided by the Higher Education Commission of Pakistan in the Computer Science Curriculum 2025. The structure ensures a balanced combination of general education, core computer science knowledge, and specialized courses in software engineering. It aims to develop strong theoretical foundations, practical computing skills, and professional competencies required for the design, development, and management of modern software systems. The program also incorporates interdisciplinary learning, professional certification, internship, and a capstone project to enhance industry readiness and support national technological advancement. The following table presents the overall credit hour distribution and course categories of the program.

Category	Category Code	Credit Hours	Courses
General Education Courses	GE (6,9)	34	14
Computer Science Core Courses	CC (1)	48	14(+6)
Software Engineering Specialization Electives	SE (2)	24	8
Interdisciplinary / Allied Courses	AC (5)	12	4
Professional Certification	PC (4)	3	—
Field Experience / Internship	FE (7)	3	—
Capstone Project / Final Year Project	FP (3)	6	—
Math Deficiency ²	MD (0)	6	2
Total		130	40

Course Coding Scheme

Category Code – YNS, where

Category Codes are GE, CC, SE, AC, PC, FE, FP, and MD for General Education Courses, Computer Science Core Courses, Software Engineering Specialization Electives Courses, Interdisciplinary / Allied Courses, Professional Certification, Field Experience / Internship, Capstone Project / Final Year Project, and Math Deficiency respectively. Y represents earliest year of offering, N is a numeric code from 0 – 9 for a course category, and S is a serial number. Code for the Lab course will be Category Code – YNS – L.

² These Math courses will be Non-Credit courses with only Pass/Fail grade assigned to the students.

General Education Courses: 34 Credits (14 Courses)

As per HEC UGE Policy V 1.1 and subsequent notifications, following courses are mandatory to be part of every undergraduate degree program, hence the same are included in the schemes of BS Computer Science (Specialization in Software Engineering).

Sr.	Code	Course Title (Category)	Prerequisite	Cr. Hrs.
1.	GE-263	Professional Practices (Arts & Humanities)		2 (2,0)
2.	GE-262	Applied Physics (Natural Science)		3 (2,1)
3.	GE-163	Business Management (Social Science)		2 (2,0)
4.	GE-162	Functional English		3 (3,0)
5.	GE-165	Expository Writing		3 (3,0)
6.	GE-161	Discrete Structures (Quantitative Reasoning)		3 (3,0)
7.	GE-164	Probability and Statistics (Quantitative Reasoning)		3 (3,0)
8.	GE-264	Islamic Studies / Ethics		2 (2,0)
9.	GE-261	Ideology and Constitution of Pakistan		2 (2,0)
10.	GE-160	Application of Information & Communication Technologies		3 (2,1)
11.	GE-265	Entrepreneurship		2 (2,0)
12.	GE-260	Civics and Community Management		2 (2,0)
13.	GE-166	Pakistan Studies		2 (2,0)
14.	GE-167	Fehm-e-Quran – I		1 (0,1)
15.	GE-168	Fehm-e-Quran – II		1 (0,1)
Total				34 (30, 4)

Computer Science Core Courses: 48 Credit Hours (20 Courses)

The following table lists the Computer Science Core Courses that constitute the mandatory major component of the BS Computer Science (Specialization in Software Engineering) program as per the HEC Computer Science Curriculum 2025. These courses provide a comprehensive foundation in essential computer science concepts and practical skills, covering fundamental programming, data management, systems design, algorithms, networking, and emerging technologies. Completion of all core courses is a degree requirement and ensures that students build the necessary competencies for advanced study and professional practice in software engineering and related fields.

Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Cr. Hrs.
1.	CC-112	Programming Fundamentals		3 (3,0)
2.	CC-112-L	Programming Fundamentals Lab		1 (0,1)
3.	CC-211	Object Oriented Programming	Programming Fundamentals	3 (3,0)
4.	CC-211-L	Object Oriented Programming Lab	Programming Fundamentals	1 (0,1)
5.	CC-215	Database Systems		3 (3,0)
6.	CC-215-L	Database Systems Lab		1 (0,1)
7.	CC-110	Digital Logic Design		3 (3,0)
8.	CC-110-L	Digital Logic Design Lab		1 (0,1)
9.	CC-213	Data Structures	Object Oriented Programming	3 (3,0)
10.	CC-213-L	Data Structures Lab	Object Oriented Programming	1 (0,1)
11.	CC-312	Information Security		3 (3,0)
12.	CC-310	Artificial Intelligence		3 (3,0)
13.	CC-214	Computer Networks		3 (3,0)
14.	CC-212	Software Engineering		3 (3,0)
15.	CC-210	Computer Organization and Architecture	Digital Logic Design	3 (2,1)
16.	CC-311	Operating Systems		3 (3,0)
17.	CC-311-L	Operating Systems Lab		1 (0,1)
18.	CC-313	Design and Analysis of Algorithms	Data Structures	3 (3,0)
19.	CC-211	Theory of Automata		3 (3,0)
20.	CC-312	Cloud Computing		3 (3,0)
Total				48 (41, 7)

Software Engineering Specialization Electives Courses: 24 Credit Hours (08 Courses)

The following table presents the Software Engineering Specialization Elective Courses, which form the elective component of the Software Engineering track within the BS Computer Science (Specialization in Software Engineering) program. As per the HEC Computer Science Curriculum 2025, students must select courses totaling 24 credit hours from this list to satisfy the specialization requirements. These electives are designed to provide advanced knowledge and practical skills in software engineering, enabling graduates to meet professional and industry standards.

To maintain alignment with evolving industry trends, technological advancements, and emerging software engineering needs, the list of specialization electives is extendable. The department may introduce new courses or modify existing ones, ensuring that graduates remain competitive and well-prepared for the dynamic software industry.

Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Cr. Hrs.
1.	SE-221	Software Requirement Engineering	Software Engineering	3 (3,0)
2.	SE-426	Software Project Management		3 (3,0)
3.	SE-325	Software Quality and Testing	Software Engineering	3 (2,1)
4.	SE-421	DevOps Principles and Practices		3 (3,0)
5.	SE-422	Software Re-Engineering		3 (3,0)
6.	SE-222	Software Design and Architecture	Software Requirement Engineering	3 (3,0)
7.	SE-323	Parallel and Distributed Computing	Data Structures	3 (3,0)
8.	SE-324	Software Construction and Development	Object Oriented Programming	3 (3,0)
9.	SE-322	Human Computer Interaction		3 (3,0)
10.	SE-424	Formal Methods and Design		3 (3,0)
11.	SE-423	Cross-platform Application Development	Object Oriented Programming	3 (3,0)
12.	SE-425	Advanced Topics in Software Engineering	Software Engineering	3 (3,0)
13.	SE-321	Machine Learning		3 (2,1)
14.	SE-426	Internet of Things and Its Applications	Computer Networks	3 (2,1)
15.	SE-427	Green Software Engineering	Software Engineering	3 (3,0)
16.	SE-325	Mobile Application Development	Object Oriented Programming	3 (3,0)

Interdisciplinary / Allied Courses: Min 12 Credits

As per HEC UGE Policy V 1.1, interdisciplinary courses of Min 12 credit hours are required in four-year BS Computer Science (Specialization in Software Engineering) program, to complement holistic understanding of the major. These courses can be selected from the following list which is further extendable. However, first two courses in the below mentioned list are mandatory as decided by the National Curriculum Revision Committee (NCRC).

Sr.	Interdisciplinary / Allied Courses	Cr. Hrs.
1.	Calculus and Analytical Geometry (<i>Mandatory</i>)	3
2.	Linear Algebra (<i>Mandatory</i>)	3
3.	Discrete Structure	3
4.	Probability & Statistics	3
5.	Economics	3
6.	Organizational Behavior	3
7.	Media and Information Literacy	3
8.	E-Governance and ICT Policy	3
9.	Environmental Science	3
10.	Professional Practice	3
11.	Digital Marketing & E-Commerce	3
12.	Technology Management	3
13.	Financial Technology (FinTech) Fundamentals	3
14.	Geographic Information Systems (GIS) & Spatial Data	3
15.	Bioinformatics	3
16.	Smart Cities & Urban Informatics	3
17.	Cyber Law & Digital Policy	3
18.	Technology and Sustainable Development	3
19.	DevOps Principles and Practices	3
20.	Software Project Management	3

Professional Certification: 03 Credit Hours

BS Computer Science (Specialization in Software Engineering) students are required to complete professional certification(s) (equivalent to 3 credit hours in total) over the period of four-year program as a mandatory condition for degree completion in accordance with “Revised Curriculum for Degree Programs in Computer Science” by Higher Education Commission (HEC), Government of Pakistan, diary No.: 8-36/HEC/Acad/2025 (Curriculum), dated: October 14, 2025.

Field Experience / Internship: 03 Credit Hours

Internships of three (03) credit hours must be completed in accordance with HEC Undergraduate Education Policy V 1.1. in accordance with “Revised Curriculum for Degree Programs in Computer Science” by Higher Education Commission (HEC), Government of Pakistan, diary No.: 8-36/HEC/Acad/2025 (Curriculum), dated: October 14, 2025. The Field Experience may be offered during summer/winter semester.

Capstone Project / Final Year Project: 06 Credit Hours

The Final Year Project may be split in 7th and 8th Semester.

Math Deficiency: 06 Credit Hours (Non-Credited)

These Math courses will be non-credit courses with only Pass/Fail grade assigned to the students.

Sr.	Code	Course Title	Prerequisite	Cr. Hrs.
1.	MD-001	Math Deficiency - I		3 (3,0)
2.	MD-002	Math Deficiency - II		3 (3,0)

Scheme of Studies:

This section presents the Scheme of Studies for the BS Computer Science (Software Engineering Specialization) program, outlining the sequence and distribution of courses across each semester. It provides a structured overview of core, specialization, interdisciplinary, and general education courses, along with credit hours and prerequisites. The scheme is designed to ensure a balanced progression of knowledge and skills, enabling students to achieve the Program Learning Outcomes (PLOs) and Program Educational Objectives (PEOs) while gaining both theoretical foundations and practical competencies in software engineering.

Semester - I						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	GE-160	Applications of ICT		GE	3	(2,1)
2.	GE-262	Applied Physics		GE	3	(2,1)
3.	CC-110	Digital Logic Design		CC	3	(3,0)
4.	CC-110-L	Digital Logic Design Lab		CC	1	(0,1)
5.	GE-264	Islamic Studies / Ethics		GE	2	(2,0)
6.	GE-261	Ideology and Constitution of Pakistan		GE	2	(2,0)
7.	GE-162	Functional English		GE	3	(3,0)
8.	GE-167	Fehm-e-Quran – I		GE	1	(0,1)
9.	MD-001	Math Deficiency – I *		MD	3*	(3,0)*
Credit Hours (Semester - I)					18	(14,4)
Semester – II						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	CC-112	Programming Fundamentals		CC	3	(3,0)
2.	CC-112-L	Programming Fundamentals Lab		CC	1	(0,1)
3.	CC-210	Computer Organization and Architecture	CC-110: Digital Logic Design	CC	3	(2,1)
4.	GE-161	Discrete Structures (Quantitative Reasoning)		GE	3	(3,0)
5.	GE-165	Expository Writing		GE	3	(3,0)
6.	GE-166	Pakistan Studies		GE	2	(2,0)
7.	GE-168	Fehm-e-Quran – II		GE	1	(0,1)
8.	MD-002	Math Deficiency – II *		MD	3*	(3,0)*
Credit Hours (Semester - II)					16	(13,3)

Semester – III						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	CC-211	Object Oriented Programming	CC-211: Programming Fundamentals	CC	3	(3,0)
2.	CC-211-L	Object Oriented Programming Lab	CC-211: Programming Fundamentals	CC	1	(0,1)
3.	CC-215	Database Systems		CC	3	(3,0)
4.	CC-215-L	Database Systems Lab		CC	1	(0,1)
5.	CC-212	Software Engineering		CC	3	(3,0)
6.	CC-211	Theory of Automata		CC	3	(3,0)
7.	AC-152	Calculus and Analytical Geometry		AC	3	(3,0)
Credit Hours (Semester - III)					17	(15,2)
Semester – IV						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	CC-213	Data Structures	CC-211: Object Oriented Programming	CC	3	(3,0)
2.	CC-213-L	Data Structures Lab	CC-211: Object Oriented Programming	CC	1	(0,1)
3.	CC-214	Computer Networks		CC	3	(3,0)
4.	SE-221	Software Requirement Engineering	CC-212: Software Engineering	SE	3	(3,0)
5.	CC-310	Artificial Intelligence		CC	3	(3,0)
6.	SE-325	Mobile Application Development	CC-211: Object Oriented Programming	SE	3	(3,0)
7.	AC-	Linear Algebra		AC	3	(3,0)
Credit Hours (Semester - IV)					19	(18,1)
Semester – V						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	CC-311	Operating Systems		CC	3	(3,0)
2.	CC-311	Operating Systems Lab		CC	1	(0,1)
3.	SE-222	Software Design and Architecture	SE-221: Software Requirement Engineering	SE	3	(3,0)
4.	SE-324	Software Construction and Development	CC-211: Object Oriented Programming	SE	3	(3,0)
5.	CC-313	Design and Analysis of Algorithms	CC-213: Data Structures	CC	3	(3,0)
6.	GE-164	Probability and Statistics		GE	3	(3,0)
7.	GE-163	Business Management (Social Science)		GE	2	(2,0)
Credit Hours (Semester - V)					18	(17,1)
Semester – VI						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	CC-312	Cloud Computing		CC	3	(3,0)
2.	CC-312	Information Security		CC	3	(3,0)
3.	SE-325	Software Quality and Testing	CC-212: Software Engineering	SE	3	(3,0)
4.	SE-322	Human Computer Interaction		SE	3	(3,0)
5.	SE-321	Machine Learning		SE	3	(2,1)
6.	GE-265	Entrepreneurship		GE	2	(2,0)
Credit Hours (Semester - VI)					17	(17,0)

Semester – VII						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	FP-431	Final Year Project – I		FP	3	(0,3)
2.	SE-423	Cross-platform Application Development	CC-211: Object Oriented Programming	SE	3	(3,0)
3.	PC-X4X	Professional Certification(s)		PC	3	(3,0)
4.	GE-263	Professional Practices		GE	2	(2,0)
5.	GE-260	Civics and Community Management		GE	2	(2,0)
Credit Hours (Semester - VII)					13	(10,3)
Semester – VIII						
Sr.	Code	Course Title	Pre-Requisite/Co-Requisite	Domain	Cr. Hrs.	
1.	FP-432	Final Year Project - II	Final Year Project - I	FP	3	(0,3)
2.	AC-	DevOps Principles and Practices		AC	3	(3,0)
3.	AC-	Software Project Management		AC	3	(3,0)
4.	FE-X7X	Field Experience / Internship		FE	3	(3,0)
Credit Hours (Semester - VIII)					12	(09,3)
Total Credit Hours					130	(115,15)

Computer Science Core Courses: 48 Credit Hours (20 Courses)

This section details the Course Learning Outcomes (CLOs), their mapping to Program Learning Outcomes (PLOs), and the associated Bloom's Taxonomy levels, along with the outlines of the core courses in Computer Science.

Programming Fundamentals			
CLOs	1. Demonstrate proficiency in writing, debugging, and executing basic programs using programming languages such as C or Python. 2. Explain core programming concepts including variables, control structures, functions, and data types. 3. Develop simple algorithms to solve computational problems. 4. Apply best practices in coding to produce efficient and readable programs. 5. Analyze program outputs and troubleshoot common errors effectively.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Program writing & debugging	PLO2, PLO5
	CLO2	Conceptual understanding	PLO1, PLO2
	CLO3	Algorithm development	PLO2, PLO3
	CLO4	Coding best practices	PLO5, PLO7
	CLO5	Output analysis & debugging	PLO3, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Apply (C3)	Writing & executing programs
	CLO2	Understand (C2)	Explaining concepts
	CLO3	Apply (C3)	Algorithm development
	CLO4	Apply → Analyze (C3–C4)	Code quality & efficiency
	CLO5	Analyze (C4)	Debugging & error analysis

Course Title	Programming Fundamentals
Course Code	CC-112
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	CC-211: Object Oriented Programming
Course Introduction	This course provides fundamental concepts of programming to freshmen. The course is a prerequisite to many other courses, therefore, students are strongly advised to cover all contents and try to achieve CLOs to the maximum possible level.
Course Description	Introduction to Problem Solving, Algorithms, Programming, and C Language: Problem Solving, a brief review of Von-Neumann Architecture., The C Programming Language, Pseudo-code, Concept of Variable, Data types in Pseudo-code, The C Standard Library and Open Source, Input/Output, Arithmetic expressions, Assignment statement, Operator precedence, Concept of Integer division, Flowchart and its notations, Typical C Program Development Environment, Role of Compiler and Linker, Test Driving C Application. Introduction to C Programming: A Simple C Program: Printing Text, Adding Two Integer, Memory Concepts, Arithmetic in C, Operators. Decision Making: Equality and Relational Operators. Structured Program Development: The if, if...else, while Nested Control Statements. Program Control: for, switch, do...while, break, continue, Logical Operators. Functions: Modularizing Program in C, Math Library Functions, Function Definitions and Prototypes, Function-Call Stack and Stack Frames, Stack rolling and unrolling, Headers, Passing Arguments by Value and by Reference, Random Number Generation, Scope Rules, Recursion, Recursion vs Iteration. Arrays: Defining Arrays, Character Arrays, Static and Automatic Local Arrays, Passing Arrays to Function, Sorting and Searching Arrays, Multidimensional and Variable Length Arrays. Pointers: Pointer Definitions and Initialization, Pointer Operators, Passing Arguments to Function by Reference, Using the const and sizeof Operator, Pointer Expressions and Arithmetic, Pointers and Arrays, Array of Pointers, Function Pointers. Characters and Strings: Strings and Characters, Character Handling Library, String Functions, Library Functions. Formatted Input/Output: Streams, Formatted Output with printf, Formatted Input with scanf. Structures: Defining Structures, Accessing Structure Member, Structures and Functions, typedef, Unions. Bit Manipulation and Enumeration: Bitwise Operators, Bit Fields, Enumeration Constants. File Processing: Files and Streams, Creating, Reading and Writing data to a Sequential and a Random-Access File. Preprocessor: #include, #define, Conditional Compilation, #error and #pragma, # and ## Operators, Predefined Symbolic Constants, Assertions. Other Topics: Variable Length Argument List, Using Command Line Arguments, Compiling Multiple-Source-File Programs, Program Termination with exit and atexit, Suffixes for Integer and Floating-Point Literals, Signal Handling, Dynamic Memory Allocation calloc and realloc, goto. Advance Topics: Self-Referential Structures, Linked Lists. Efficiency of Algorithms, Selection and Insertion Sort.
Text Book(s)	Paul Deitel, Harvey Deitel, C How To Program, 9th Edition, Pearson, 2022.
Reference Material	1. Tony Gaddis, Starting out with Programming Logic and Design, 5th Edition, Pearson, 2018. 2. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie 3. Object Oriented Programming in C++ by Robert Lafore 1. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

Title	Programming Fundamentals Lab
--------------	-------------------------------------

Code	CC-112-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	CC-211: Object Oriented Programming
Course Introduction	This course provides fundamental concepts of programming to freshmen. The course 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.
Course Description	Implementation and Practice of the concepts studied in “CC-112 Programming Fundamentals” Introduction to Problem Solving, Algorithms, Programming, and C Language: Problem Solving, a brief review of Von-Neumann Architecture., The C Programming Language, Pseudo-code, Concept of Variable, Data types in Pseudo-code, The C Standard Library and Open Source, Input/Output, Arithmetic expressions, Assignment statement, Operator precedence, Concept of Integer division, Flowchart and its notations, Typical C Program Development Environment, Role of Compiler and Linker, Test Driving C Application. Introduction to C Programming: A Simple C Program: Printing Text, Adding Two Integer, Memory Concepts, Arithmetic in C, Operators. Decision Making: Equality and Relational Operators. Structured Program Development: The if, if...else, while Nested Control Statements. Program Control: for, switch, do...while, break, continue, Logical Operators. Functions: Modularizing Program in C, Math Library Functions, Function Definitions and Prototypes, Function-Call Stack and Stack Frames, Stack rolling and unrolling, Headers, Passing Arguments by Value and by Reference, Random Number Generation, Scope Rules, Recursion, Recursion vs Iteration. Arrays: Defining Arrays, Character Arrays, Static and Automatic Local Arrays, Passing Arrays to Function, Sorting and Searching Arrays, Multidimensional and Variable Length Arrays. Pointers: Pointer Definitions and Initialization, Pointer Operators, Passing Arguments to Function by Reference, Using the const and sizeof Operator, Pointer Expressions and Arithmetic, Pointers and Arrays, Array of Pointers, Function Pointers. Characters and Strings: Strings and Characters, Character Handling Library, String Functions, Library Functions. Formatted Input/Output: Streams, Formatted Output with printf, Formatted Input with scanf. Structures: Defining Structures, Accessing Structure Member, Structures and Functions, typedef, Unions. Bit Manipulation and Enumeration: Bitwise Operators, Bit Fields, Enumeration Constants. File Processing: Files and Streams, Creating, Reading and Writing data to a Sequential and a Random-Access File. Preprocessor: #include, #define, Conditional Compilation, #error and #pragma, # and ## Operators, Predefined Symbolic Constants, Assertions. Other Topics: Variable Length Argument List, Using Command Line Arguments, Compiling Multiple-Source-File Programs, Program Termination with exit and atexit, Suffixes for Integer and Floating-Point Literals, Signal Handling, Dynamic Memory Allocation calloc and realloc, goto. Advance Topics: Self-Referential Structures, Linked Lists. Efficiency of Algorithms, Selection and Insertion Sort.
Text Book(s)	A. Paul Deitel, Harvey Deitel, C How To Program, 9th Edition, Pearson, 2022.
Reference Material	1. Tony Gaddis, Starting out with Programming Logic and Design, 5th Edition, Pearson, 2018. 2. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie 3. Object Oriented Programming in C++ by Robert Lafore 4. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

Object Oriented Programming			
CLOs	- Understand and apply the principles of object-oriented programming, including encapsulation, inheritance, and polymorphism. - Design and implement classes and objects to model real-world entities. - Write reusable and modular code using OOP concepts. - Analyze the advantages of OOP over procedural programming paradigms. - Develop small-scale applications utilizing OOP concepts in languages like Java or C++.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Apply OOP principles (encapsulation, inheritance, polymorphism)	PLO1, PLO2
	CLO2	Design and implement classes & objects	PLO2, PLO4
	CLO3	Write reusable & modular code	PLO2, PLO4, PLO5
	CLO4	Analyze OOP vs procedural paradigms	PLO2, PLO3
	CLO5	Develop small applications using OOP	PLO2, PLO4, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand → Apply (C2–C3)	Applying OOP principles (encapsulation, inheritance, polymorphism)
	CLO2	Apply → Create (C3–C6)	Designing and implementing classes & objects
	CLO3	Apply → Create (C3–C6)	Writing reusable and modular code using OOP concepts
	CLO4	Analyze → Evaluate (C4–C5)	Analyzing OOP vs procedural programming paradigms
	CLO5	Apply → Create (C3–C6)	Developing small applications using OOP concepts

Title	Object Oriented Programming
Code	CC-211
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	CC-112 Programming Fundamentals
Co-Requisite	None
Follow-up	CC-213 Data Structures
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.
Course Description	Introduction to Object-oriented Design, History and Advantages of Object-oriented Design. Introduction to OOP and C++: Brief description of C++ concepts, Introduction to OOP. Introduction to Classes Objects and Member Functions: Encapsulation and Abstraction, Class and Object, Getter/Setter Functions, Access Specifiers, Constructors, Overloaded Constructor, Default Constructor, Destructor. Functions: Inline Functions, Function Overloading. Class Templates array: Function Templates, Class Templates array, Vectors and Multidimensional Array, Reference to private Data Members, Default Member wise Assignment, const Objects, const Member Functions. Composition and Aggregation: Object Composition and Aggregation, Class Separation using header. Friend Classes and Functions: Friend Functions, Friend Classes. static Members: “this” pointer, static Data Members, static Member Functions. Copy Constructor: Default Copy Constructor. Operator Overloading: Overloaded Operators of Standard Library, Operator Overloading, Overloading Binary Operators, Overloading Unary Operators, Overloading ++ Operator, Overloading – Operator, Dynamic Memory Management, Operators as Members vs Non-Members, Conversion between Types, Explicit Constructor and Conversion Operators, Overloading the Function call Operator. Stream I/O: Introduction, Streams, Streams Input, Streams Output, Object Streams, data and object serialization using object streams. Inheritance: Introduction, Base and Derived Classes, Relationships between Base and Derived Classes, Constructors in Derived Classes, Destructor in Derived Classes, public protected and private Inheritance. Polymorphism: Relationship among Objects in Inheritance, Virtual Functions, Virtual Destructors, Pure Virtual Functions, Abstract and Concrete Classes. File Processing: Files and Streams, create a Sequential File, read a Sequential File, update a Sequential File, Random Access File, create a Random-Access File, read a Random-Access File, update a Random-Access File, Exception Handling: Flow of Control, Rethrowing an Exception, Constructor Destructor and Exception handling. Generic Programming Concepts: Custom Templates, Class Templates, Function Templates, Arguments to Templates, Overloading Function Templates. Standard Library: Containers, Iterators, Adapters, Sequence Containers, Associative Containers, Container Adapters, Minimum Iterator Requirements, Lambda Expressions, Function Objects.
Text Book(s)	A. P. Deitel, H. Deitel, C++ How To Program, 10th Edition, Pearson.
Reference Material	1. Robert Lafore, Object Oriented Programming in C++, 3rd Edition. 2. Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, Pearson, 2018. 3. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

Title	Object Oriented Programming Lab
Code	CC-211-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	CC-112 Programming Fundamentals
Co-Requisite	None
Follow-up	CC-213 Data Structures
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.
Course Description	Implementation and Practice of the concepts studied in “CC-211 Object Oriented Programming” Introduction to Object-oriented Design, History and Advantages of Object-oriented Design. Introduction to OOP and C++: Brief description of C++ concepts, Introduction to OOP. Introduction to Classes Objects and Member Functions: Encapsulation and Abstraction, Class and Object, Getter/Setter Functions, Access Specifiers, Constructors, Overloaded Constructor, Default Constructor, Destructor. Functions: Inline Functions, Function Overloading. Class Templates array: Function Templates, Class Templates array, Vectors and Multidimensional Array, Reference to private Data Members, Default Member wise Assignment, const Objects, const Member Functions. Composition and Aggregation: Object Composition and Aggregation, Class Separation using header. Friend Classes and Functions: Friend Functions, Friend Classes. static Members: “this” pointer, static Data Members, static Member Functions. Copy Constructor: Default Copy Constructor. Operator Overloading: Overloaded Operators of Standard Library, Operator Overloading, Overloading Binary Operators, Overloading Unary Operators, Overloading ++ Operator, Overloading – Operator, Dynamic Memory Management, Operators as Members vs Non-Members, Conversion between Types, Explicit Constructor and Conversion Operators, Overloading the Function call Operator. Stream I/O: Introduction, Streams, Streams Input, Streams Output, Object Streams, data and object serialization using object streams. Inheritance: Introduction, Base and Derived Classes, Relationships between Base and Derived Classes, Constructors in Derived Classes, Destructor in Derived Classes, public protected and private Inheritance. Polymorphism: Relationship among Objects in Inheritance, Virtual Functions, Virtual Destructors, Pure Virtual Functions, Abstract and Concrete Classes. File Processing: Files and Streams, create a Sequential File, read a Sequential File, update a Sequential File, Random Access File, create a Random-Access File, read a Random-Access File, update a Random-Access File, Exception Handling: Flow of Control, Rethrowing an Exception, Constructor Destructor and Exception handling. Generic Programming Concepts: Custom Templates, Class Templates, Function Templates, Arguments to Templates, Overloading Function Templates. Standard Library: Containers, Iterators, Adapters, Sequence Containers, Associative Containers, Container Adapters, Minimum Iterator Requirements, Lambda Expressions, Function Objects.
Text Book(s)	A. P. Deitel, H. Deitel, C++ How To Program, 10th Edition, Pearson.
Reference Material	1. Robert Lafore, Object Oriented Programming in C++, 3rd Edition. 2. Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, Pearson, 2018. 3. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

Database Systems			
CLOs	1. Design and normalize relational database schemas based on user requirements. 2. Write SQL queries for data retrieval, insertion, update, and deletion. 3. Explain the concepts of database transactions, concurrency, and recovery. 4. Implement basic database management tasks using popular database management systems. 5. Analyze the role of databases in information systems and their security considerations.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Design and normalize relational database schemas	PLO2, PLO4
	CLO2	Write SQL queries for data retrieval, insertion, update, deletion	PLO2, PLO5
	CLO3	Explain database transactions, concurrency, and recovery	PLO1, PLO2
	CLO4	Implement basic database management tasks using DBMS	PLO2, PLO4, PLO5
	CLO5	Analyze database role in information systems & security	PLO2, PLO3, PLO8
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Apply → Create (C3–C6)	Designing and normalizing relational database schemas
	CLO2	Apply (C3)	Writing SQL queries for data retrieval, insertion, update, and deletion
	CLO3	Understand → Analyze (C2–C4)	Explaining database transactions, concurrency, and recovery
	CLO4	Apply → Create (C3–C6)	Implementing basic DBMS tasks using modern database tools
	CLO5	Analyze → Evaluate (C4–C5)	Analyzing database roles in information systems & security considerations

Course Title	Database Systems
Course Code	CC-215
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	
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
Course Description	File Systems and Databases: Introduction, A File system Critique, Database Systems, Database approach vs file-based system, database architecture, three level schema architecture, data independence, Database Models. Introduction to RDBMS: Logical view of Data; Entities and Attributes, Tables and their Characteristics, Keys; relational data model, attributes, schemas, tuples, domains, relation instances, keys of relations, integrity constraints. Relational Algebra: Relational Database Operators, selection, projection, Cartesian product, types of joins. Entity Relationship (E-R) Modeling: Basic Modeling Concepts, entity sets, attributes, relationship, entity-relationship diagrams, Normalization of Database Tables: Objectives, Forms, Normalization and Database Design, functional dependencies, normal forms, Denormalization, Structured Query Language (SQL): Introduction, DDL Commands, Joins and subqueries in SQL, Grouping and aggregation in SQL, DML Commands, DCL Commands, Complex Queries and SQL Functions, Procedural SQL; Triggers, Stored procedures. Database Design: The System Development Life Cycle (SDLC), The Database Life Cycle (DBLC), Database Design Strategies, Transaction Management and Concurrency Control: Introduction, Transaction Properties and Types, Concurrency Control Issues, Database Recovery Management. DDBMS: Evolution, Components, Distributed processing and distributed databases, Distributed database transparency features. Distributed database design, Data fragmentation, Data replication, NoSQL systems.
Text Book(s)	1. Carlos Coronel, Steven Morris, Database Systems: Design, Implementation & Management, 13 th Edition, Cengage Learning, 2017. ISBN-10: 1337627909.
Reference Material	1. Jeffrey A. Hoffer, Ramesh Venkataraman, Heikki Topi, Modern Database Management, 12 th Edition, Pearson, 2015. ISBN-10: 0133544613. 2. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management, 6 th Edition, Pearson, 2015. ISBN-10: 1292061189. 3. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7 th Edition, Pearson, 2016. ISBN-10: 1292097612.

Course Title	Database Systems Lab
Course Code	CC-215-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Introduction	The course aims to introduce the Structured Query Language (SQL). It covers the set of commands related to Data Retrieval, Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL). It will follow up by procedural flavor of SQL (PL/SQL).
Course Description	Introduction to SQL environment: Writing Basic SQL Statements; SELECT Statement: Arithmetic Expressions, Operator Precedence, Null Value, Column Alias, Concatenation Operator, FROM Clause: Table list, Table Alias. Restricting and Sorting Data; WHERE Clause: Comparison operators, Logical operators, ORDER BY clause, Display table Structure. Single row functions: character functions, number functions, date functions, type conversion functions. Multi row Functions: Sum, Average, Standard deviation, Variance, Subgrouping of data, Group by Clause, use of Having clause. Join: Cross product, natural join, Equi-join, Non equi-join, left outer-join, right outer-join, self-join. Subquery: use of subquery, subquery syntax, Multiple Column Subqueries, pairwise comparison, Non-pair wise comparison, Null Value in a subquery, Subquery in From Clause. Creating and Altering SQL tables: Create table statement, Defining Constraints, Column Level and Table Level, NOT NULL Constraint, UNIQUE Key Constraint, PRIMARY Key Constraint, FOREIGN Key Constraint, CHECK Constraint, Alter table statement, drop statement, Data Manipulation: Insert, Update, Delete statements. SQL Objects: Views, Sequences, Indexes. User Management: Create user, user privileges, user groups, Grant, Revoke statements. Introduction to Procedural SQL (PL/SQL), Sections of a PL/SQL block, Variable declaration and initialization, SELECT statement in PL/SQL, Arithmetic expressions, Selection, Repetition, Exception Handling, Cursors, Stored Procedures and Functions, Introduction to Database Triggers
Text Book(s)	1. Introduction to Oracle 9i: SQL 2. Michael McLaughlin, Oracle Database 11g PL/SQL Programming, 1st Edition, McGraw-Hill Education, 2008, ISBN: 0071494456.
Reference Material	1. Jason Price, Oracle Database 11g SQL, McGraw Hill. ISBN: 0071498508.

Digital Logic Design			
CLOs	1. Draw and interpret digital logic diagrams, including combinational and sequential circuits. 2. Design basic digital components such as multiplexers, flip-flops, and encoders. 3. Analyze the behavior of digital systems using truth tables and Boolean algebra. 4. Implement simple digital circuits using logic gates. 5. Understand the fundamentals of digital system design and their applications.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Draw & interpret digital logic diagrams	PLO2, PLO3
	CLO2	Design basic digital components (multiplexers, flip-flops, encoders)	PLO2, PLO4
	CLO3	Analyze behavior of digital systems using truth tables & Boolean algebra	PLO2, PLO3
	CLO4	Implement simple digital circuits using logic gates	PLO2, PLO4, PLO5
	CLO5	Understand fundamentals of digital system design & applications	PLO1, PLO2
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand → Apply (C2–C3)	Drawing & interpreting digital logic diagrams
	CLO2	Apply → Create (C3–C6)	Designing basic digital components (multiplexers, flip-flops, encoders)
	CLO3	Analyze (C4)	Analyzing behavior of digital systems using truth tables & Boolean algebra
	CLO4	Apply → Create (C3–C6)	Implementing simple digital circuits using logic gates
	CLO5	Understand → Analyze (C2–C4)	Understanding fundamentals of digital system design & applications

Course Title	Digital Logic Design
Course Code	CC-110
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	CC-210 Computer Organization and Architecture
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.
Course Description	Topics: Introduction to Digital Systems, Number Systems, Introduction to Boolean Algebra, Basic theorems and properties of Boolean Algebra, Boolean Functions, Logic Gates, NAND and NOR Implementation, Representation of Function in Sum of Minterms or Product of Maxterms, Simplification of Boolean function using Karnaugh Map, Don't care Conditions, The Tabulation Method, Introduction to Combinational Logic, Design of Adders, Design of Subtractors, Code Convertors, Analysis Procedure of Combinational Circuits, Binary Parallel Adders, Decimal Adders, Magnitude Comparator, Decoders and its applications, Multiplexers, Demultiplexers, Encoders, ROM, Programmable Logic Array (PLA), Introduction to Sequential Circuits, Basic Flip Flop, Clocked RS Flip Flop, Clocked D Flip Flop, Clocked JK Flip Flop, Clocked T Flip Flop, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Flip Flop Excitation tables, Design Procedure, Design of Counters, Design with State Equations, Introduction to Registers, Shift Registers, Ripple Counters, Synchronous Counters, Timing Sequences, Memory Unit, Random Access Memory. Introduction Programmable Logic Devices (CPLD, FPGA), Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim. Familiarization with Digital Electronic Trainer, Logic gates operations, Half Adder Operation, Full Adder Operation, Half Subtractor Operation, Full Adder Operation, 7-Segment Display Operation, Decoder Operation, BCD To 7-Segment Display, Multiplexer Operation, Using Multiplexer and Demultiplexer / Decoder, Multiplexing 7-Segment Displays, Comparator Operations, D Latch and Flip-Flop Operation, Latching BCD Data for Displaying On 7-Segment Display, JK Flip-Flop Operation, Random Access Memories
Text Book(s)	M. Morris Mano, Digital Logic and Computer Design, 1 st Edition, Pearson, 1979, ISBN: 0132145103.
Reference Material	2. Thomas L. Floyd, Digital Fundamentals, 10th Edition, Prentice Hall, 2008, ISBN: 0132359235. 3. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e

Course Title	Digital Logic Design Lab
Course Code	CC-110-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	CC-210 Computer Organization and Architecture
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.
Course Description	Implementation and Practice of the concepts studied in “CC-110 Digital Logic Design” Topics: Introduction to Digital Systems, Number Systems, Introduction to Boolean Algebra, Basic theorems and properties of Boolean Algebra, Boolean Functions, Logic Gates, NAND and NOR Implementation, Representation of Function in Sum of Minterms or Product of Maxterms, Simplification of Boolean function using Karnaugh Map, Don't care Conditions, The Tabulation Method, Introduction to Combinational Logic, Design of Adders, Design of Subtractors, Code Convertors, Analysis Procedure of Combinational Circuits, Binary Parallel Adders, Decimal Adders, Magnitude Comparator, Decoders and its applications, Multiplexers, Demultiplexers, Encoders, ROM, Programmable Logic Array (PLA), Introduction to Sequential Circuits, Basic Flip Flop, Clocked RS Flip Flop, Clocked D Flip Flop, Clocked JK Flip Flop, Clocked T Flip Flop, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Flip Flop Excitation tables, Design Procedure, Design of Counters, Design with State Equations, Introduction to Registers, Shift Registers, Ripple Counters, Synchronous Counters, Timing Sequences, Memory Unit, Random Access Memory. Introduction Programmable Logic Devices (CPLD, FPGA), Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim. Familiarization with Digital Electronic Trainer, Logic gates operations, Half Adder Operation, Full Adder Operation, Half Subtractor Operation, Full Adder Operation, 7-Segment Display Operation, Decoder Operation, BCD To 7-Segment Display, Multiplexer Operation, Using Multiplexer and Demultiplexer / Decoder, Multiplexing 7-Segment Displays, Comparator Operations, D Latch and Flip-Flop Operation, Latching BCD Data for Displaying On 7-Segment Display, JK Flip-Flop Operation, Random Access Memories
Text Book(s)	M. Morris Mano, Digital Logic and Computer Design, 1 st Edition, Pearson, 1979, ISBN: 0132145103.
Reference Material	4. Thomas L. Floyd, Digital Fundamentals, 10th Edition, Prentice Hall, 2008, ISBN: 0132359235. 5. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e

Data Structures			
CLOs	1. Implement and analyze various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. 2. Select appropriate data structures to optimize algorithm performance. 3. Demonstrate proficiency in traversing and manipulating data structures. 4. Evaluate the efficiency of algorithms based on data structure choices. 5. Solve complex problems using suitable data structures and algorithms.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Implement & analyze arrays, linked lists, stacks, queues, trees, graphs	PLO2, PLO3
	CLO2	Select appropriate data structures to optimize algorithms	PLO2, PLO3, PLO4
	CLO3	Traverse & manipulate data structures efficiently	PLO2, PLO5
	CLO4	Evaluate efficiency of algorithms based on data structures	PLO2, PLO3, PLO4
	CLO5	Solve complex problems using suitable data structures & algorithms	PLO2, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Apply → Analyze (C3–C4)	Implementing and analyzing arrays, linked lists, stacks, queues, trees, graphs
	CLO2	Analyze → Evaluate (C4–C5)	Selecting appropriate data structures to optimize algorithm performance
	CLO3	Apply → Create (C3–C6)	Traversing and manipulating data structures efficiently
	CLO4	Analyze → Evaluate (C4–C5)	Evaluating efficiency of algorithms based on data structure choices
	CLO5	Apply → Create (C3–C6)	Solving complex problems using suitable data structures and algorithms

Title	Data Structures
Code	CC-213
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	CC-211 Object Oriented Programming
Co-Requisite	None
Follow-up	
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.
Course Description	Algorithm Specification: Properties of Algorithm, examples, performance, analysis, measurement, and Big Oh notation. Introduction to ADTs: Array and Polynomial as an ADT, Sparse Matrices, and Representation of Arrays. The Stack ADT: Linked list and array implementations, Expressions, Postfix Notation, and Infix to postfix conversion. The Queue ADT: Linked and array implementations of circular and double ended queue. Recursion: Recursive Definition and Processes, Writing Recursive Programs. Divide and Conquer Algorithms, Self-Referencing Classes and Dynamic Memory Allocation, Garbage Collection. Linked List: Singly Linked Lists, Circular Lists, Linked Stacks and Queues (Double Ended List), Doubly Linked Lists. Trees: Introduction to Trees, Logical construction and Traversing of Binary Trees, Implementation of Binary Trees (Insertion and Traversing), Searching and deletion in Binary Trees, Binary Search Tree, Introduction to Balanced and AVL Trees. Heaps: Heaps and Heaps as Priority Queues, Double Ended Priority Queue. Searching: Linear Search, Binary Search, and Types of Indexing. Hashing: Hash Functions: Division, Open Addressing; Overflow Handling: Chaining; Introduction to advanced topics: B-Trees, M-Way Trees, Generalized List etc. Sorting: Selection, Insertion, Merge, Quick, Bubble, Heap, Shell, Radix, and Bucket sorts. Graphs: Graph terminology, Adjacency List and Adjacency Matrix and Adjacency list representation of Graph; Elementary Graph Operations: Breadth First Search and Depth First Search, Spanning Trees (BFSST, DFSST), topological order, shortest path.
Text Book(s)	A. Ellis Horowitz, Sartaj Sahni, and D. Mehta, "Fundamentals of Data Structures in C++", 2 nd Ed., Computer Science Press
Reference Material	1. Adam B. Drozdek, Data Structure and Algorithm in C++, 4th Ed., Cengage Learning 2. Mark Allen Weiss, "Data Structure and Algorithms in C++", 2nd Ed., Pearson Education 3. D. Malhotra and N. Malhotra. Data Structures and Program Design Using C++. 4. Tenenbaum, M. Augenstein, and Y. Lang Sam, "Data Structures using C and C++" 2nd Ed., Prentice Hall

Title	Data Structures Lab
Code	CC-213-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	CC-211 Object Oriented Programming
Co-Requisite	None
Follow-up	
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.
Course Description	Implementation and Practice of the concepts studied in “CC-213 Data Structures” Algorithm Specification: Properties of Algorithm, examples, performance, analysis, measurement, and Big Oh notation. Introduction to ADTs: Array and Polynomial as an ADT, Sparse Matrices, and Representation of Arrays. The Stack ADT: Linked list and array implementations, Expressions, Postfix Notation, and Infix to postfix conversion. The Queue ADT: Linked and array implementations of circular and double ended queue. Recursion: Recursive Definition and Processes, Writing Recursive Programs. Divide and Conquer Algorithms, Self-Referencing Classes and Dynamic Memory Allocation, Garbage Collection. Linked List: Singly Linked Lists, Circular Lists, Linked Stacks and Queues (Double Ended List), Doubly Linked Lists. Trees: Introduction to Trees, Logical construction and Traversing of Binary Trees, Implementation of Binary Trees (Insertion and Traversing), Searching and deletion in Binary Trees, Binary Search Tree, Introduction to Balanced and AVL Trees. Heaps: Heaps and Heaps as Priority Queues, Double Ended Priority Queue. Searching: Linear Search, Binary Search, and Types of Indexing. Hashing: Hash Functions: Division, Open Addressing; Overflow Handling: Chaining; Introduction to advanced topics: B-Trees, M-Way Trees, Generalized List etc. Sorting: Selection, Insertion, Merge, Quick, Bubble, Heap, Shell, Radix, and Bucket sorts. Graphs: Graph terminology, Adjacency List and Adjacency Matrix and Adjacency list representation of Graph; Elementary Graph Operations: Breadth First Search and Depth First Search, Spanning Trees (BFSST, DFSST), topological order, shortest path.
Text Book(s)	A. Ellis Horowitz, Sartaj Sahni, and D. Mehta, “Fundamentals of Data Structures in C++”, 2 nd Ed., Computer Science Press
Reference Material	1. Adam B. Drozdek, Data Structure and Algorithm in C++, 4th Ed., Cengage Learning 2. Mark Allen Weiss, “Data Structure and Algorithms in C++”, 2nd Ed., Pearson Education 3. D. Malhotra and N. Malhotra. Data Structures and Program Design Using C++. 4. Tenenbaum, M. Augenstein, and Y. Lang Sam, “Data Structures using C and C++” 2nd Ed., Prentice Hall

Information Security			
CLOs	1. Explain fundamental concepts of information security, including confidentiality, integrity, and availability. 2. Identify common security threats and vulnerabilities. 3. Apply security measures such as encryption, authentication, and access control. 4. Conduct basic security audits and risk assessments. 5. Understand legal and ethical issues related to information security.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain fundamental concepts of information security	PLO1, PLO2
	CLO2	Identify common security threats & vulnerabilities	PLO2, PLO3
	CLO3	Apply security measures (encryption, authentication, access control)	PLO2, PLO4, PLO5
	CLO4	Conduct basic security audits & risk assessments	PLO2, PLO3, PLO4
	CLO5	Understand legal & ethical issues in information security	PLO8, PLO9
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining fundamental security concepts (confidentiality, integrity, availability)
	CLO2	Analyze (C4)	Identifying security threats and vulnerabilities
	CLO3	Apply → Create (C3–C6)	Implementing security measures like encryption and authentication
	CLO4	Analyze → Evaluate (C4–C5)	Conducting security audits and risk assessments
	CLO5	Understand → Evaluate (C2–C5)	Understanding legal and ethical issues related to information security

Course Title	Information Security
Course Code	CC-312
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	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.
Course Description	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.
Text Book(s)	1. M. Whitman and H. Mattord, Principles of Information Security, 6th edition.
Reference Material	1. William Stallings, Computer Security: Principles and Practice, 3rd edition. 2. Dieter Gollmann, Computer Security, 3rd edition. 3. William Easttom, Computer Security Fundamentals, 3rd edition.

Artificial Intelligence			
CLOs	1. Describe core AI concepts including search algorithms, knowledge representation, and reasoning. 2. Implement basic AI algorithms for problem-solving and decision-making. 3. Analyze the applications of AI in real-world scenarios. 4. Discuss ethical considerations and limitations of AI systems. 5. Develop simple AI models using appropriate tools and frameworks		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Describe core AI concepts (search, knowledge representation, reasoning)	PLO1, PLO2
	CLO2	Implement basic AI algorithms for problem-solving	PLO2, PLO4, PLO5
	CLO3	Analyze applications of AI in real-world scenarios	PLO2, PLO3
	CLO4	Discuss ethical considerations and limitations of AI systems	PLO8, PLO9
	CLO5	Develop simple AI models using appropriate tools & frameworks	PLO2, PLO4, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Describing core AI concepts (search, knowledge representation, reasoning)
	CLO2	Apply → Create (C3–C6)	Implementing basic AI algorithms for problem-solving
	CLO3	Analyze (C4)	Analyzing AI applications in real-world scenarios
	CLO4	Understand → Evaluate (C2–C5)	Discussing ethical considerations and limitations of AI systems
	CLO5	Apply → Create (C3–C6)	Developing simple AI models using tools and frameworks

Course Title	Artificial Intelligence
Course Code	CC-310
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	
Co-Requisite	
Follow-up	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.
Course Description	An Introduction to Artificial Intelligence and its applications towards Knowledge Based Systems; Symbolic AI: the physical symbol system hypothesis. Search: exhaustive & heuristic search techniques. 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, Game playing, Genetic algorithms, Introduction to Machine Learning for AI, Decision Trees, Bayesian classification, Artificial Neural Networks, Computer Vision. Introduction to Python programming, Logic programming: knowledge representation & search in the context of logic programming. Reasoning in logic programming: unification, horn clause logic, and resolution, Knowledge Representation Schemas: Logic, frames, semantic nets, scripts; problems in knowledge representation. Expert systems.
Text Book(s)	1. Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 3rd edition, Prentice Hall, Inc., 2010.
Reference Material	1. Luger, G.F. and Stubblefield, W.A., 2009. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley. 2. George F. Luger, Artificial Intelligence - Structures and Strategies for Complex Problem Solving, 6th Edition, Pearson, 2008, ISBN-13: 978-0321545893. 3. Hart, P.E., Stork, D.G. and Duda, R.O., Pattern classification. John Willey & Sons, 2001. 4. Ivan Bratko, Prolog: Programming for Artificial Intelligence, 4th Edition, Pearson, 2011, ISBN-13: 978-0321417466. 5. P. Winston, Artificial Intelligence, 3rd Edition, Pearson, 1992, ISBN-13: 978-0201533774.

Computer Networks			
CLOs	1. Explain the fundamental concepts of computer networking, including protocols, topologies, and models (OSI, TCP/IP). 2. Configure and troubleshoot basic network devices and connections. 3. Analyze network security threats and mitigation techniques. 4. Demonstrate understanding of data transmission and error handling. 5. Design simple network architectures to meet organizational needs.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain fundamental networking concepts (protocols, topologies, OSI/TCP-IP)	PLO1, PLO2
	CLO2	Configure and troubleshoot network devices & connections	PLO2, PLO5
	CLO3	Analyze network security threats & mitigation techniques	PLO2, PLO3, PLO8
	CLO4	Demonstrate understanding of data transmission & error handling	PLO2, PLO3
	CLO5	Design simple network architectures to meet organizational needs	PLO2, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining fundamental networking concepts (protocols, topologies, OSI/TCP-IP)
	CLO2	Apply → Create (C3–C6)	Configuring and troubleshooting network devices & connections
	CLO3	Analyze → Evaluate (C4–C5)	Analyzing network security threats & mitigation techniques
	CLO4	Understand → Analyze (C2–C4)	Understanding data transmission & error handling
	CLO5	Apply → Create (C3–C6)	Designing simple network architectures to meet organizational needs

Course Title	Computer Networks
Course Code	CC-214
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	None
Course Description	Introduction: 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, subnetting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks. The lab contents are divided into two parts. After completion of the first part, the students will be able to understand Computer Networks basics, network types, layered communication models and protocols. The contents have been organized in such a way as to form the base for learning the concepts to be taught in the next part. The assimilation of the networking concepts will enable the students to apply them for solving practical problems. After completion of this part, the students will be familiar with networking, routing, switching, and Setting up of networks from scratch, major protocols involved in communication and their configurations. Students will also be made aware of the state of the art areas in case they would like to pursue this course in future.
Text Book(s)	1. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 6th Edition, Pearson, 2012, ISBN: 0132856204. 2. T. Lammle, CCNA Cisco Certified Network Associate Deluxe Study Guide, 6th Edition, Sybex, 2011, ISBN: 978-0-470-90108-3.
Reference Material	1. Andrew S. Tanenbaum, David J. Wetherall, Computer Networks, 5th Edition, Prentice Hall, 2010, ISBN: 9332518742. 2. William Stallings, Data and Computer Communications, 10th Edition, Pearson, 2013, ISBN: 0133506487. 3. Behrouz A. Forouzan, Data Communication and Computer Networks, 5th Edition, McGraw-Hill, 2012, ISBN: 0073376221. 4. R. Perlman, Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2nd Edition, Addison-Wesley, 1999, ISBN: 0201634481.

Software Engineering			
CLOs	1. Apply software development life cycle models to manage projects effectively. 2. Develop software requirements specifications and design documents. 3. Implement and test software applications using best practices. 4. Analyze and manage software project risks and quality. 5. Collaborate effectively in team-based software projects.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Apply software development life cycle models	PLO2, PLO4
	CLO2	Develop software requirements & design documents	PLO2, PLO4
	CLO3	Implement and test software applications	PLO2, PLO4, PLO5
	CLO4	Analyze and manage software project risks & quality	PLO2, PLO3, PLO4
	CLO5	Collaborate effectively in team-based software projects	PLO2, PLO6
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Apply → Analyze (C3–C4)	Applying software development life cycle models to projects
	CLO2	Apply → Create (C3–C6)	Developing software requirements specifications and design documents
	CLO3	Apply → Create (C3–C6)	Implementing and testing software applications using best practices
	CLO4	Analyze → Evaluate (C4–C5)	Analyzing and managing software project risks and quality
	CLO5	Apply → Analyze (C3–C4)	Collaborating effectively in team-based software projects

Course Title	Software Engineering
Course Code	CC-212
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow Up	None
Course Description	Introduction: Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software Process Models: Waterfall Model, Incremental Model, Prototyping Model, Spiral Model, RAD Model. Agile Software Development: Agile process models, Agile development techniques. Introduction to Project Management, Introduction to Requirements Engineering, Functional and non-functional requirements. Analysis Model: Context models, Interaction models, Structural models, behavioral models, model driven engineering, Data modeling, Functional Modeling, Behavioral Modeling. Software Design: Data Design, Architectural Design, Component Level Design, User Interface Design. Object Oriented Analysis & Design Basics: Introduction to UML, UML Diagrams. Use Case Modeling, Rational Rose overview, Use case modeling in Rational Rose. Domain Model: Identifying business classes, Domain Model Associations, Domain Model Attributes, Implementation of Sequence Diagram and Domain model in Rational Rose. Interaction Diagram: Sequence diagrams, Collaboration Diagrams, Implementation of Sequence and Collaboration diagrams in Rational Rose. Design Class Diagram, Mapping Design to Code. Software Testing Fundamentals. Design patterns, Software testing and quality assurance. Software evolution. Project Management: Project planning, configuration management. Software Process improvement.
Text Book(s)	1. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015, ISBN-13: 978-0133943030.
Reference Material	1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2nd Edition, Pearson, 2002, ISBN-13: 978-0133056990.

Computer Organization and Architecture			
CLOs	1. Describe the structure and function of computer components such as CPU, memory, and I/O devices. 2. Write and understand basic assembly language programs. 3. Analyze how hardware components interact during program execution. 4. Explain the concepts of instruction set architecture and microarchitecture. 5. Optimize programs considering hardware limitations.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Describe structure & function of CPU, memory, I/O devices	PLO1, PLO2
	CLO2	Write & understand basic assembly language programs	PLO2, PLO5
	CLO3	Analyze hardware interaction during program execution	PLO2, PLO3
	CLO4	Explain instruction set architecture & microarchitecture	PLO1, PLO2
	CLO5	Optimize programs considering hardware limitations	PLO2, PLO3, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Describing structure & function of CPU, memory, I/O devices
	CLO2	Apply (C3)	Writing and understanding basic assembly language programs
	CLO3	Analyze (C4)	Analyzing hardware interactions during program execution
	CLO4	Understand → Analyze (C2–C4)	Explaining instruction set architecture and microarchitecture
	CLO5	Apply → Analyze (C3–C4)	Optimizing programs considering hardware limitations

Course Title	Computer Organization and Architecture
Course Code	CC-210
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	CC-110 Digital Logic Design
Co-Requisite	None
Follow-up	None
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.
Course Description	Topics: Introduction to computer systems, Evolution of Intel Microprocessor, Introduction to Assembly Language, Computer Organization, The Components of a Microcomputer System, Instruction Cycle, Memory Architecture, Memory Representation & Hierarchy, Data, Address, Control Busses, Intel 8086 family of Microprocessors, Organization of Intel 8088/8086 Processor, Registers and their categories Function of Registers, Memory Addressing, Real Mode Memory Structure, Memory Segmentation (Segment/Offset Scheme), Computer Instructions for Basic computer (Memory Reference, Register Reference and I/O instructions), Addressing modes, Instruction Cycle, Timing and Decoding, RTL of the instructions, Complete flow chart for the Basic Computer Operation, Addressing Modes, Design of the CPU of a basic computer Assembly Language Syntax, Program data, Variables, Variables, Program Structure, Memory Models, Data Segments, Stack Segment, Code Segment, Variants of MOV instruction, Some Basic Instructions, XCHG, ADD, SUB, INC, DEC, NEG, Input and Output Instructions, The Processor Status and Flags Register, Flow Control Instructions, Unconditional Jump, Various Conditional Jumps, Looping Structures, Logic Instructions, AND, OR, XOR, NOT, TEST, Shift Instructions, Rotate Instructions, Procedures to Input Binary, Decimal, Hexadecimal Numbers, Procedures to output Binary, Decimal, Hexadecimal Numbers, The Stack, PUSH and POP Instructions, CALL and RET instructions, MUL instruction, DIV instruction, Related Programming examples, XLAT instruction, String Instructions, MOVS/MB, MOVSW, MOVSB/W, STOSB/W, SCASB/W, CMPSB/W, File Operations, Reading a File, Writing a File
Text Book(s)	1. Charles Marut, Ytha Yu, Assembly Language Programming and Organization of the IBM PC, 1st Edition, McGraw-Hill, 1992, ISBN: 9780070726925. 2. M. Morris Mano, Computer System Architecture, 3rd Edition, Pearson, 1993, ISBN: 9780131755635.
Reference Material	1. Barry B. Brey, The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium and Pentium Pro-Processor, Pentium II, Pentium III, Pentium 4", 7th Edition, Prentice Hall, 2005, ISBN: 0131195069. 2. Kip R. Irvine, Assembly Language for Intel Based Computers, 4th Edition, Prentice Hall, 2002, ISBN: 9780130910134.

Operating Systems			
CLOs	1. Explain the functions and services provided by operating systems. 2. Manage processes, threads, and synchronization mechanisms. 3. Analyze memory management techniques and file systems. 4. Implement basic scheduling algorithms. 5. Evaluate operating system performance and security features.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain OS functions & services	PLO1, PLO2
	CLO2	Manage processes, threads & synchronization	PLO2, PLO3, PLO5
	CLO3	Analyze memory management & file systems	PLO2, PLO3, PLO4
	CLO4	Implement basic scheduling algorithms	PLO2, PLO5
	CLO5	Evaluate OS performance & security features	PLO2, PLO3, PLO8
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining OS functions and services
	CLO2	Apply → Analyze (C3–C4)	Managing processes, threads, and synchronization mechanisms
	CLO3	Analyze (C4)	Analyzing memory management techniques and file systems
	CLO4	Apply (C3)	Implementing basic scheduling algorithms
	CLO5	Analyze → Evaluate (C4–C5)	Evaluating OS performance and security features

Course Title	Operating Systems
Course Code	CC-311
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Introduction	To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.
Course Description	Introduction: Operating systems basics, system calls, process concept and scheduling, inter-process communication, multithreaded programming, multithreading models, threading issues. Process Scheduling: Algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks. Memory Management: swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files. File Systems: file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management. System Protection: Virtual machines, operating system security. Interacting with Linux Operating System: Virtualization and Hypervisors. File System Architecture: Schematic view of a standard UNIX file system. File System Mounting: Introduction to the concept of file system mounting. Linux configuration files related to file system mounting. File Permissions: standard file permissions. Use of chmod and chown commands. Device files: Seven File Types in Linux and the concept of device files. Terminal Attributes: Overview of Terminal Devices and current attributes of the terminal driver. Hard and Soft Links. Managing services using systemd: Introduction to Linux system daemon. Shell commands to manage services using systemctl. Booting process of a Linux system. Linux System Programming: Linux System Call Interface, Use of GNU gcc compiler Process Creation and Termination: getpid(), getppid(), fork(), exit(), wait() and execl() system calls. File management in Linux. Concept of PPFDT. Concept of input, output and error redirection. Inter Process Communication: Linux IPC tools, Pipes, FIFOs and Sockets. Use of pipes and fifos on a Linux terminal. Signals: Signal delivery and execution of a signal handler. Synchronous and Asynchronous signals. Threads and Scheduling: Writing multi-threaded C programs using library calls from the POSIX pthread library like pthread_create(), pthread_join(), and pthread_exit().
Text Book(s)	1. A. Silberschatz, P. B. Galvin, G. Gagne, Operating Systems Concepts, 9 th Edition, Wiley, 2012, ISBN: 1118063333.
Reference Material	1. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 4th Edition, Pearson, 2014, ISBN: 013359162X. 2. William Stallings, Operating Systems: Internals and Design Principles, 9th Edition, Pearson, 2017, ISBN: 0134670957.

Course Title	Operating Systems Lab
Course Code	CC-311-L
Credit Hours	1 (0,1)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Introduction	To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.
Course Description	Implementation and Practice of the concepts studied in “CC-311 Operating Systems” Introduction: Operating systems basics, system calls, process concept and scheduling, inter-process communication, multithreaded programming, multithreading models, threading issues. Process Scheduling: Algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks. Memory Management: swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files. File Systems: file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management. System Protection: Virtual machines, operating system security. Interacting with Linux Operating System: Virtualization and Hypervisors. File System Architecture: Schematic view of a standard UNIX file system. File System Mounting: Introduction to the concept of file system mounting. Linux configuration files related to file system mounting. File Permissions: standard file permissions. Use of chmod and chown commands. Device files: Seven File Types in Linux and the concept of device files. Terminal Attributes: Overview of Terminal Devices and current attributes of the terminal driver. Hard and Soft Links. Managing services using systemd: Introduction to Linux system daemon. Shell commands to manage services using systemctl. Booting process of a Linux system. Linux System Programming: Linux System Call Interface, Use of GNU gcc compiler Process Creation and Termination: getpid(), getppid(), fork(), exit(), wait() and execl() system calls. File management in Linux. Concept of PPFDT. Concept of input, output and error redirection. Inter Process Communication: Linux IPC tools, Pipes, FIFOs and Sockets. Use of pipes and fifos on a Linux terminal. Signals: Signal delivery and execution of a signal handler. Synchronous and Asynchronous signals. Threads and Scheduling: Writing multi-threaded C programs using library calls from the POSIX pthread library like pthread_create(), pthread_join(), and pthread_exit().
Text Book(s)	2. A. Silberschatz, P. B. Galvin, G. Gagne, Operating Systems Concepts, 9 th Edition, Wiley, 2012, ISBN: 1118063333.
Reference Material	3. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 4th Edition, Pearson, 2014, ISBN: 013359162X. 4. William Stallings, Operating Systems: Internals and Design Principles, 9th Edition, Pearson, 2017, ISBN: 0134670957.

Design and Analysis of Algorithms			
CLOs	1. Design efficient algorithms for common computational problems. 2. Analyze algorithm complexity using Big O notation. 3. Solve problems involving recursion, divide-and-conquer, and dynamic programming. 4. Compare different algorithmic approaches to problem-solving. 5. Demonstrate correctness and optimality of algorithms.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Design efficient algorithms for common computational problems	PLO2, PLO4
	CLO2	Analyze algorithm complexity using Big O notation	PLO2, PLO3
	CLO3	Solve problems using recursion, divide-and-conquer, dynamic programming	PLO2, PLO4, PLO5
	CLO4	Compare different algorithmic approaches to problem-solving	PLO2, PLO3, PLO4
	CLO5	Demonstrate correctness & optimality of algorithms	PLO2, PLO4, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Apply → Create (C3–C6)	Designing efficient algorithms for computational problems
	CLO2	Analyze (C4)	Analyzing algorithm complexity using Big O notation
	CLO3	Apply → Analyze (C3–C4)	Solving problems using recursion, divide-and-conquer, and dynamic programming
	CLO4	Analyze → Evaluate (C4–C5)	Comparing different algorithmic approaches to problem-solving
	CLO5	Analyze → Evaluate (C4–C5)	Demonstrating correctness and optimality of algorithms

Course Title	Design and Analysis of Algorithms
Course Code	CC-313
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	CC-213 Data Structures
Co-Requisite	
Follow Up	None
Course Introduction	Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.
Course Description	Topics: Role of Algorithm in Computing, Analysis on nature of input and size of input , Designing Algorithms, Growth of Functions, Asymptotic Notations, Big-O, Big Ω , Big Θ , little-o, little- ω , loop invariants, Brute Force Approach, Divide-and-conquer approach; Sorting Algorithm analysis, Merge, Quick Sort, Greedy approach; Recursion and recurrence relations, Time Complexity of Recursive Algorithms, Algorithm Design Techniques, Dynamic Programming, Greedy Algorithms, String Matching, Search trees; Heaps; Hashing; Graph algorithms, sparse graphs, DFS, BFS, Minimum Spanning Trees, Shortest Path Algorithms, NP Completeness, Polynomial Time Algorithm, Polynomial Time verification.
Text Book(s)	1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 3 rd Edition, The MIT Press, 2009, ISBN-10: 0262033844, ISBN-13: 978-0262033848
Reference Material	1. Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos 2. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne

Theory of Automata			
CLOs	1. Explain the concepts of finite automata, regular expressions, and formal languages. 2. Design automata to recognize specific languages. 3. Demonstrate the equivalence of automata, regular expressions, and grammars. 4. Analyze the limitations of finite automata and context-free grammars. 5. Apply automata theory to compiler design and language processing.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain finite automata, regular expressions & formal languages	PLO1, PLO2
	CLO2	Design automata to recognize specific languages	PLO2, PLO4
	CLO3	Demonstrate equivalence of automata, regular expressions & grammars	PLO2, PLO3
	CLO4	Analyze limitations of finite automata & context-free grammars	PLO2, PLO3, PLO4
	CLO5	Apply automata theory to compiler design & language processing	PLO2, PLO4, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining finite automata, regular expressions & formal languages
	CLO2	Apply → Create (C3–C6)	Designing automata to recognize specific languages
	CLO3	Apply → Analyze (C3–C4)	Demonstrating equivalence of automata, regular expressions & grammars
	CLO4	Analyze (C4)	Analyzing limitations of finite automata & context-free grammars
	CLO5	Apply → Analyze (C3–C4)	Applying automata theory to compiler design & language processing

Course Title	Theory of Automata
Course Code	CC-211
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow Up	None
Course Description	Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output), Pumping lemma and non-regular language Grammars and PDA: CFGs, Derivations, derivation trees and ambiguity, Simplifying CFLs, Normal form grammars and parsing, Decidability, Context sensitive languages, grammars and linear bounded automata (LBA), Chomsky's hierarchy of grammars Turing Machines Theory: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Defining Computers by TMs.
Suggested Instructional/ Reading Material	1. Daniel I. A. Cohen, "Introduction to computer theory", 2nd Ed., Wiley India Pvt. Limited, 2007. ISBN 0471137723, 9780471137726 2. Peter Linz, "An Introduction to Formal Languages and Automata", 5th Ed., Jones & Bartlett Publishers, 2011. ISBN: 144961552X, 9781449615529 3. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011 4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, 2005, New Age Publishers

Cloud Computing			
CLOs	1. Describe fundamental cloud computing models and services (IaaS, PaaS, SaaS). 2. Deploy and manage applications in cloud environments. 3. Analyze the benefits and challenges of cloud computing. 4. Implement basic cloud security and compliance measures. 5. Evaluate cloud solutions for scalability, cost, and performance.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Describe cloud computing models & services	PLO1, PLO2
	CLO2	Deploy & manage cloud applications	PLO2, PLO5
	CLO3	Analyze cloud benefits & challenges	PLO2, PLO3
	CLO4	Implement cloud security & compliance	PLO2, PLO4, PLO8
	CLO5	Evaluate cloud solutions (scalability, cost, performance)	PLO2, PLO3, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Describing fundamental cloud computing models & services
	CLO2	Apply → Create (C3–C6)	Deploying and managing applications in cloud environments
	CLO3	Analyze (C4)	Analyzing the benefits and challenges of cloud computing
	CLO4	Apply → Analyze (C3–C4)	Implementing cloud security and compliance measures
	CLO5	Analyze → Evaluate (C4–C5)	Evaluating cloud solutions for scalability, cost, and performance

Course Title	Cloud Computing
Course Code	CC-312
Credit Hours	3 (3,0)
Category	Computer Science Core
Prerequisite	None
Co-Requisite	None
Follow-up	None
Course Introduction	Nowadays cloud computing becomes popular for individuals' users as well as for the business community. It provides on-demand huge computation and storage resources which is attractive for a large number of users mainly due to pay-per-usage charging model. This course will introduce topics related to Cloud Computing and also expose students to the latest tools and technologies used in Cloud Computing. The aim of this course is to provide an in-depth knowledge of Cloud Computing topics. The course would also expose students to important methods and tools used Cloud Computing.
Course Description	Introduction To Cloud Computing, Introduction To Cloud Computing (Cont.), AWS Services, EC2 hands, Accessing AWS S3 , Virtualization , Containerization/Dockers, Scalable Web Application in The Cloud, Scalable Web Application in The Cloud (Cont.), Load Balancing AWS ELB, Nginx, Ipvadm, Ldirectord, Web Application Deployment Models, Introduction to Big Data, Introduction to Big Data (Cont.), Introduction to NoSQL, MongoDB and DynamoDB, Revision, MapReduce, Distributed Systems, Consistency and Fault Tolerance in Distributed Systems, Big Data Transformation Methods , Blockchain, Practical Considerations in Cloud Computing, Future of the Cloud Computing and Big Data
Text Book(s)	1. JothyRosenberg and Arthur Mateos;The Cloud at Your Service;Manning Publications. ISBN: 1935182528 2. Paul Zikopoulos and Chris Eaton; Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data; McGraw-Hill. ISBN: 0071790535
Reference Material	1. Kyle Banker, Peter Bakkum, Shaun Verch, Douglas Garrett, and Tim Hawkins; MongoDB in Action, Second Edition. 2. Clinton W. Brownley; Foundations for Analytics with Python from Non-Programmer to Hacker.

Software Engineering Specialization Electives Courses: 24 Credit Hours (08 Courses)

This section presents the Course Learning Outcomes (CLOs), their mapping to Program Learning Outcomes (PLOs), and alignment with Bloom's Taxonomy, along with the course outlines for the Software Engineering specialization courses. The listed courses represent the core and elective offerings for the specialization, and the list can be further extended as needed to reflect evolving industry requirements and advancements in software engineering practices.

Software Requirement Engineering			
CLOs	1. Explain the principles of requirements elicitation, analysis, and specification. 2. Apply structured techniques to document software requirements. 3. Analyze stakeholder requirements to identify conflicts and derive feasible solutions.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain requirements elicitation, analysis & specification	PLO1, PLO2
	CLO2	Apply techniques to document software requirements	PLO2, PLO5
	CLO3	Analyze stakeholder requirements & resolve conflicts	PLO2, PLO3, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Focuses on explaining core concepts of requirements engineering
	CLO2	Apply (C3)	Involves applying structured techniques for documentation
	CLO3	Analyze (C4)	Requires analyzing stakeholder needs and resolving requirement conflicts

Course Title	Software Requirement Engineering
Course Code	SE-221
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	CC-212 Software Engineering
Co-Requisite	None
Follow Up	
Course Description	Introduction to Requirements Engineering, Software Requirements, classification of requirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements Management Problems , Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.
Text Book(s)	1. Software Requirements, Wiegers K. &Beatty J., 3rd Ed. Microsoft Press, 2013
Reference Material	1. Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick. 3rd Ed, Springer-Verlag London Limited, 2011. 2. Requirements Engineering and Management for Software Development Projects, Chemuturi M., Springer New York, 2013.

Software Project Management			
CLOs	1. Explain fundamental project management concepts. 2. Apply project planning techniques to software projects. 3. Analyze risks and constraints in project execution. 4. Use project management tools to plan and track progress. 5. Demonstrate leadership and teamwork in simulated project environments.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain fundamental project management concepts	PLO1, PLO2
	CLO2	Apply project planning techniques to software projects	PLO2, PLO5
	CLO3	Analyze risks and constraints in project execution	PLO2, PLO3, PLO4
	CLO4	Use project management tools to plan and track progress	PLO5
	CLO5	Demonstrate leadership and teamwork in project environments	PLO9, PLO10
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining core project management concepts
	CLO2	Apply (C3)	Applying planning techniques to real software projects
	CLO3	Analyze (C4)	Analyzing risks and constraints in project execution
	CLO4	Apply (C3)	Using tools for planning and tracking project progress
	CLO5	Affective (A4: Characterize)	Demonstrating leadership and teamwork in project settings

Course Title	Software Project Management
Course Code	SE-426
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	None
Follow Up	None
Course Introduction	This course provides an introduction to the key concepts, methods, and best practices needed to effectively manage software projects. Through a combination of lectures, case studies, and interactive discussions, students will gain insights into the entire software development lifecycle, from ideation to deployment, while emphasizing time management, budgeting, stakeholder communication, and risk management.
Course Description	Introduction: Project Management. The Project Management Process Groups. Scope triangle, The S curve. Five Phases of Project Management Life Cycle: Defining, planning, executing, controlling, closing, Scope of project, Project Work Breakdown Structure (WBS), Project Time Management. Estimate Activity Duration: Causes of variation in activity duration, five methods of Estimating Activity Duration. Project Networks: Critical Path Method (CPM), Build the project network, Analysis of the project network, Project Evaluation Review Technique (PERT), Activity on Arrow, Activity on Node, GANTT Chart, Using MS-Project to draw GANTT chart and project Networks. Project Proposal: Purpose of the project proposal, Contents of the project proposal, Format of the Project Proposal. Staffing and Personnel Planning: Personnel Plan, Team structures, Democratic decentralized, Controlled Decentralized, Controlled Centralized, Coordination and communication issues, Maslow's need theory of motivation, Software Project Team Roles and Responsibilities, Risk Identification, Analysis and Management. Project Management: Integration Management. Scope Management. Cost Management. Quality Management. Human Resource Management. Communications Management. Risk Management. Procurement Management. Management Tools. Configuration Management: Change Management and Control, baselines, Version Control, Earned Value Analysis for Project Monitoring and Control. Examples of Earned Value Analysis, Project Quality Assurance Plans, SQA Process Project Quality Standards, Project Documents.
Text Book(s)	1. Guide to the Project Management Body of Knowledge (PMBOK® Guide), 6 th Edition, Project Management Institute, 2017, ISBN-10: 1628251840.
Reference Material	1. Teresa Luckey, Joseph Phillips, Software Project Management for Dummies, 1st Edition, For Dummies Publisher, 2006, ISBN 0471749346.

Software Quality and Testing			
CLOs	1. Explain software testing and quality assurance principles. 2. Apply the concepts of software verification and validation. 3. Apply various software testing techniques in development project. 4. Design automated test cases.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain software testing and quality assurance principles	PLO1, PLO2
	CLO2	Apply software verification and validation concepts	PLO2, PLO5
	CLO3	Apply various software testing techniques in development	PLO2, PLO3, PLO5
	CLO4	Design automated test cases	PLO3, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining fundamental testing and quality assurance concepts
	CLO2	Apply (C3)	Applying verification and validation concepts
	CLO3	Apply (C3)	Applying various software testing techniques
	CLO4	Create / Design (C5)	Designing automated test cases for software projects

Course Title	Software Quality and Testing
Course Code	SE-325
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	CC-212 Software Engineering
Co-Requisite	None
Follow Up	None
Course Introduction	This course is designed to introduce students to the core principles, methods, and practices of Software Quality Assurance. Through a blend of theory and hands-on exercises, learners will gain a deeper understanding of what it takes to ensure software products meet the required standards of quality before they reach the end-user.
Course Description	Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing
Text Book/s	1. Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed. CRC Press, Taylor and Francis Group, 2015
Reference Material	1. Bernard Homes, Fundamentals of Software Testing, ISTE, Wiley, 2012 2. Software Engineering, "Ian Sommerville, 9th Edition, Addison Wesley, 2011

DevOps Principles and Practices			
CLOs	1. Explain the core principles and cultural shifts necessary for successful DevOps adoption. 2. Implement automated pipelines for continuous integration and continuous delivery of software. 3. Use industry-standard tools and techniques to build and deploy modern, cloud-based applications. 4. Use collaborative tools and scripting languages to foster teamwork in DevOps practices.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain core principles and cultural shifts for DevOps adoption	PLO1, PLO2
	CLO2	Implement automated CI/CD pipelines	PLO2, PLO5
	CLO3	Use tools and techniques to build and deploy cloud-based applications	PLO2, PLO3, PLO5
	CLO4	Use collaborative tools and scripting to foster teamwork in DevOps	PLO9, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining core principles and cultural aspects of DevOps
	CLO2	Apply (C3)	Implementing automated CI/CD pipelines
	CLO3	Apply (C3)	Using tools and techniques to build and deploy cloud-based apps
	CLO4	Affective (A4)	Fostering teamwork and collaboration using DevOps practices

Course Title	DevOps Principles and Practices
Course Code	SE-421
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Introduction	The DevOps Principles and Practices course provides students with a foundation in the principles, culture, and workflows necessary for modern DevOps adoption. It covers automated pipelines for continuous integration and delivery, the use of industry-standard tools, and cloud-based application deployment. Emphasis is placed on collaborative practices, scripting, and version control to enhance teamwork and operational efficiency. By the end of the course, students will be able to apply DevOps practices to real-world software projects effectively.
Course Description	Introduction to DevOps: principles, culture, and organizational change. Agile and lean practices in software development. Continuous Integration (CI) and Continuous Delivery/Deployment (CD) concepts. Building automated pipelines using Jenkins, Git, and other CI/CD tools. Containerization and orchestration with Docker and Kubernetes. Cloud-based deployment strategies (AWS, Azure, or Google Cloud). Collaborative workflows, version control, and scripting for automation. Monitoring, logging, and feedback loops in DevOps pipelines. Security, performance, and reliability considerations in DevOps. Capstone project: designing and implementing a functional CI/CD pipeline for a software application
Text Book/s	1. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, Gene Kim, Jez Humble, Patrick Debois, John Willis 2. Accelerate: The Science of Lean Software and DevOps, Nicole Forsgren, Jez Humble, Gene Kim 3. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Jez Humble & David Farley 4. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, Jennifer Davis & Katherine Daniels 5. Site Reliability Engineering: How Google Runs Production Systems, Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy
Reference Material	1. Learning Continuous Integration with Jenkins, Nikhil Pathania 2. Docker and Kubernetes: The Complete Guide, Stephen Grider 3. Python Scripting for DevOps, Jason Cannon

Software Re Engineering			
CLOs	1. Understand the concepts and technique of software re-engineering. 2. Apply re-engineering techniques to maintain and modify software. 3. Analyze maintenance issues associated with software systems. 4. Identify solutions to mitigate software maintenance issues.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Understand concepts and techniques of software re-engineering	PLO1, PLO2
	CLO2	Apply re-engineering techniques to maintain and modify software	PLO2, PLO5
	CLO3	Analyze maintenance issues in software systems	PLO2, PLO3
	CLO4	Identify solutions to mitigate software maintenance issues	PLO2, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Learning core concepts and techniques of software re-engineering
	CLO2	Apply (C3)	Applying re-engineering techniques in practical maintenance scenarios
	CLO3	Analyze (C4)	Analyzing maintenance challenges and issues in software systems
	CLO4	Understand / Evaluate (C2–C5)	Identifying and assessing solutions to mitigate maintenance issues

Course Title	Software Re-Engineering
Course Code	SE-422
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Introduction	The Software Re-Engineering course introduces students to the principles, techniques, and practices used to improve and modernize existing software systems. The course covers strategies for analyzing legacy software, identifying areas for improvement, and applying systematic approaches to restructure, refactor, and enhance code quality, maintainability, and performance. Students will explore reverse engineering, design recovery, and re-documentation techniques, along with modern tools and methods for software modernization. By the end of the course, students will be able to assess existing software systems, plan and implement improvements, and apply re-engineering practices to support long-term software evolution and sustainability.
Course Description	Salient topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.
Text Book/s	1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, 2016 2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc.2015 3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press 2000.
Reference Material	Software Reengineering: Principles and Practice, Paul C. Jorgensen Software Reengineering, William C. Wong Refactoring: Improving the Design of Existing Code, Martin Fowler Legacy System Migration: A Systems Engineering Approach, S. Knight & K. Robertson

Software Design and Architecture			
CLOs	1. Describe key design principles and architectural styles. 2. Apply modeling techniques to represent software subsystems. 3. Collaborate in teams to present design specifications.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Describe key design principles and architectural styles	PLO1, PLO2
	CLO2	Apply modeling techniques to represent software subsystems	PLO2, PLO3, PLO5
	CLO3	Collaborate in teams to present design specifications	PLO9, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Describing software design principles and architectural styles
	CLO2	Apply (C3)	Using modeling techniques to represent software subsystems
	CLO3	Affective (A4)	Demonstrating teamwork and collaboration in presenting designs

Course Title	Software Design and Architecture
Course Code	SE-222
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	SE-221 Software Requirement Engineering
Follow Up	
Course Description	Software Design Concepts, Design principles, Object-Oriented Design with UML, System design and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Design Patterns, Exploring inheritance, Interactive systems with MVC architecture, Software reuse. Architectural design issues, Software Architecture, Architectural Structures & Styles-, Architectural Patterns, Architectural & Design Qualities, Quality Tactics, Architecture documentation, Architectural Evaluation, Model driven development.
Text Book(s)	1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, 8th Ed, McGraw-Hill Education, 2015.
Reference Material	1. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath, 2nd Ed, Universities Press, India, 2014. 2. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Hassan Gomaa, Cambridge University Press, 2011. 3. Head First Design Patterns, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates, O'Reilly Media, Inc. 2004.

Parallel and Distributed Computing			
CLOs	1. Explain concepts of concurrency, synchronization, and distributed architectures. 2. Apply parallel programming libraries to solve computational problems. 3. Analyze the performance trade-offs between sequential and parallel solutions.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain concepts of concurrency, synchronization, and distributed architectures	PLO1, PLO2
	CLO2	Apply parallel programming libraries to solve computational problems	PLO2, PLO5
	CLO3	Analyze performance trade-offs between sequential and parallel solutions	PLO2, PLO3
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining core concepts of concurrency, synchronization, and distributed architectures
	CLO2	Apply (C3)	Applying parallel programming libraries to practical problems
	CLO3	Analyze (C4)	Analyzing performance trade-offs between sequential and parallel solutions

Course Title	Parallel and Distributed Computing
Course Code	SE-323
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	CC-213 Data Structures
Co-Requisite	None
Follow-up	None
Course Introduction	The demand of huge computation and storage resources has been increasing exponentially. The course provides the methods for dealing with the emerging challenges of enhancing the power of the computational resources. This course will introduce topics related to parallel and distributed computing and also expose students to the latest tools and technologies.
Course Description	Asynchronous/synchronous computation/communication, concurrency control, fault tolerance, GPU architecture and programming, heterogeneity, interconnection topologies, load balancing, memory consistency model, memory hierarchies, Message passing interface (MPI), MIMD/SIMD, multithreaded programming, parallel algorithms & architectures, parallel I/O, performance analysis and tuning, power, programming models (data parallel, task parallel, process-centric, shared/distributed memory), scalability and performance studies, scheduling, storage systems, synchronization, and tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE).
Text Book(s)	1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum and M. V. Steen, Prentice Hall, 2nd Edition, 2007
Reference Material	2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, KHwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.

Software Construction and Development			
CLOs	1. Understand abstraction, modularity, concurrency, and software design patterns. 2. Develop optimized, error-free, and reusable code using OOP constructs. 3. Design and develop a small-scale software system collaboratively.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Understand abstraction, modularity, concurrency, and design patterns	PLO1, PLO2
	CLO2	Develop optimized, error-free, and reusable code using OOP constructs	PLO2, PLO5
	CLO3	Design and develop a small-scale software system collaboratively	PLO3, PLO9, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Understanding key programming concepts and design patterns
	CLO2	Create / Design (C5)	Developing optimized and reusable code using object-oriented constructs
	CLO3	Create / Design (C5)	Designing and developing a small software system collaboratively

Course Title	Software Construction and Development
Course Code	SE-324
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	CC-211 Object Oriented Programming
Follow-up	None
Course Introduction	The demand of huge computation and storage resources has been increasing exponentially. The course provides the methods for dealing with the emerging challenges of enhancing the power of the computational resources. This course will introduce topics related to parallel and distributed computing and also expose students to the latest tools and technologies.
Course Description	Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life cycle model characteristics, Individual and team software process, Lehman's Laws, code salvaging, and configuration management. Martin Fowler's refactoring concepts and their application to small projects. Apply Michael Feathers' "legacy code" concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).
Text Book(s)	1. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall, 2008.
Reference Material	1. The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt and David Thomas, Addison-Wesley Professional, 1999. 2. Working Effectively with Legacy Code, Michael C. Feathers. Pearson Education, Prentice-Hall, 2004. 3. Refactoring: Improving the Design of Existing Code, Martin Fowler, Addison-Wesley Professional. 1999.

Human Computer Interaction			
CLOs	1. Explain fundamental principles of user-centered design. 2. Analyze user feedback to refine interface prototypes. 3. Design interactive prototypes using modern HCI tools. 4. Identify accessibility and inclusivity as essential elements of HCI.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain fundamental principles of user-centered design	PLO1, PLO2
	CLO2	Analyze user feedback to refine interface prototypes	PLO2, PLO3
	CLO3	Design interactive prototypes using modern HCI tools	PLO3, PLO5
	CLO4	Identify accessibility and inclusivity as essential elements of HCI	PLO2, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Understanding principles of user-centered design
	CLO2	Analyze (C4)	Analyzing user feedback to improve interfaces
	CLO3	Create / Design (C5)	Designing interactive prototypes using HCI tools
	CLO4	Understand / Apply (C2–C3)	Recognizing and applying accessibility and inclusivity principles

Course Title	Human Computer Interaction
Course Code	SE-322
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Description	Topics: Interaction Paradigms: Batch Processing, Time sharing, Networking, Graphical Displays, Microprocessor, WWW, Ubiquitous Computing. Human: Fitt's Law, Human Information Processor, Emotions, Psychology and Design of Interactive Systems, Accessibility. Computers: capabilities and limitations of Computer Systems, Color Models, New Interactive Systems: Speech, touch, gestures, handwriting recognition. Interactions: Models of Interaction, Frameworks, Interaction Styles, User Experience, Interaction Design. Usability: HCI in software Process, Learnability, Flexibility, Robustness. Design Rules: Principles, Standard (ISO9241), Guidelines, Heuristics (Neilsen, Norman, Shneiderman), HCI Patterns (Mobile & Web Application UI). Implementation Support: jQuery, User Interface Management System and architectural Patterns. Evaluation Techniques: expert analysis (Heuristic Evaluation), Evaluation through user participation: Empirical methods and experiment design, statistical measures, Observational Techniques. User Support: Tutorial help, documentation, Wizards, adaptive help. Cognitive Models: KLM-GOMS Task Analysis: Hierarchical Task Analysis, Knowledge based Analysis, Entity Relationship based Techniques. Dialog Design: Diagrammatic Notations: State Transition Networks, Petri Nets, state charts, flow charts. Overview of Textual Notation: Grammars, Production Rules. Computer Mediated Communication: groupware, computer mediated communication, Meeting and decision support systems, argumentation Tools, shared application and Artefacts. Issues of Groupware implementation. Ubiquitous Computing: Introduction, information and data visualization overview.
Text Book(s)	1. Alan Dix, Janet Finlay Gregory D Abowd, and Russel Beale, "Human Computer Interaction", 3rd Ed. Pearson Prentice Hall, 2004, ISBN-13: 978-0130461094. 2. Julie A Jacko and Andrew Sears, "The Human Computer Interaction Handbook," 2nd Ed., CRC, 2007, ISBN 0-8058-5870-9.
Reference Material	1. Ben Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 6th Ed., Addison Wesley, 2016, ISBN -13: 978-0134380384: 2. William Albert and Thomas Tullis, "Measuring the User Experience", Morgan Kaufmann, 2013, Paperback ISBN: 9780124157811 3. Helen Sharp, Jenny Preece and Yvonne Rogers, "Interaction Design: Beyond Human-Computer Interaction", Wiley, 2002, ISBN-13: 978-0470665763 4. http://www.ui-patterns.com 5. http://www.androidpatterns.com 6. http://www.usability.gov

Formal Methods and Design			
CLOs	1. Explain the role of formal methods in software specification and verification. 2. Apply formal mathematical methods and tools to solve elementary computation problems. 3. Use mathematics-based techniques and tools for specifying, designing, and verifying software.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain the role of formal methods in software specification and verification	PLO1, PLO2
	CLO2	Apply formal mathematical methods and tools to solve elementary computation problems	PLO2, PLO5
	CLO3	Use mathematics-based techniques and tools for specifying, designing, and verifying software	PLO2, PLO3, PLO5
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Understanding the role of formal methods in specification and verification
	CLO2	Apply (C3)	Applying formal mathematical methods and tools to computational problems
	CLO3	Apply (C3)	Using math-based techniques for software specification, design, and verification

Course Title	Formal Methods and Design
Course Code	SE-424
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Description	Introduction to the use of mathematical models for specification and validation, Finite state machine models, models of concurrent systems, verification of models, and limitations. Analyzing well-formedness (e.g. completeness, consistency, robustness, etc.), Analyzing correctness (e.g. static analysis, simulation, model checking, etc.), Formal analysis, an introduction to VDM-SL, Sets, Sequences, Composite objects, Maps, VDMSL, Comparative Formal Methods, Proofs, Introduction to Z
Text Book(s)	1. Modern Formal Methods and Applications, Hossam A. Gabbar, Springer-Verlag 2006. 2. Formal Software Development: From VDM to Java, Charatan, Quentin, and Aaron Kans. Palgrave Macmillan, 2003. 3. Understanding Z: a Specification Language and its Formal Semantics. J. M. Spivey. 1988. Cambridge University Press, New York, NY, USA.

Cross-platform Application Development			
CLOs	1. Explain the principles, architectures, and frameworks used in cross-platform application development. 2. Design and develop applications using cross-platform frameworks. 3. Utilize client-side and server-side APIs, tools, and libraries. 4. Demonstrate responsible engineering practices by designing and deploying secure, accessible, and user-centric cross-platform solutions.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain principles, architectures, and frameworks in cross-platform development	PLO1, PLO2
	CLO2	Design and develop applications using cross-platform frameworks	PLO2, PLO3, PLO5
	CLO3	Utilize client-side and server-side APIs, tools, and libraries	PLO2, PLO5
	CLO4	Demonstrate responsible engineering practices in secure, accessible, user-centric apps	PLO4, PLO9
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Understanding core principles, architectures, and frameworks
	CLO2	Create / Design (C5)	Designing and developing cross-platform applications
	CLO3	Apply (C3)	Applying APIs, tools, and libraries to implement functionality
	CLO4	Affective (A3)	Valuing responsible engineering practices in secure and accessible design

Course Title	Cross-platform Application Development
Course Code	SE-423
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Introduction	The Cross-Platform Application Development course provides students with a comprehensive understanding of developing software applications that run across multiple platforms, including mobile, desktop, and web environments. The course begins with the study of cross-platform development principles, architectures, and frameworks, exploring how platform-independent applications can be designed and executed efficiently (CLO1). Students will gain hands-on experience in designing and developing applications using cross-platform frameworks such as React Native, Flutter, and Xamarin (CLO2). The curriculum includes the integration of client-side and server-side APIs, libraries, and tools to implement core functionalities and services in cross-platform applications (CLO3). Emphasis is placed on responsible engineering practices, including security, accessibility, usability, and deployment considerations, to ensure user-centric, robust applications (CLO4).
Course Description	Topics Covered (Explicit Technical Contents): Introduction to cross-platform application development: benefits, challenges, and frameworks Software architecture patterns for cross-platform apps (MVC, MVVM, component-based design) Development frameworks: React Native, Flutter, Xamarin overview Client-side programming: UI design, event handling, state management Server-side integration: RESTful APIs, GraphQL, backend services Data management and persistence: local storage, databases, caching strategies Networking and communication protocols Security best practices in cross-platform applications (authentication, encryption) Accessibility and usability considerations Testing strategies: unit testing, integration testing, and device/platform testing Deployment strategies for multiple platforms (App Store, Play Store, web) Continuous integration and delivery for cross-platform applications Project-based development: building a secure, user-centric, multi-platform application
Text Book(s)	1. Learning React Native: Building Native Mobile Apps with JavaScript, Bonnie Eisenman 2. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart, Alessandro Biessek 3. Cross-Platform Desktop Applications: With Electron, Neutralino, and Tauri, Paul B. Jensen 4. Pro Xamarin.Forms: Cross-Platform C# Apps for iOS, Android, and Windows, Simone Alessandria, et al. 5. Mobile App Development with Ionic, Revised Edition, Chris Griffith

Advanced Topics in Software Engineering			
CLOs	1. Explain the latest trends and emerging technologies in the software development industry. 2. Apply new technologies in software engineering projects. 3. Analyze professional practices in software engineering.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain latest trends and emerging technologies in software engineering	PLO1, PLO2
	CLO2	Apply new technologies in software engineering projects	PLO2, PLO5
	CLO3	Analyze professional practices in software engineering	PLO2, PLO3
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining latest trends and emerging technologies
	CLO2	Apply (C3)	Applying new technologies in projects
	CLO3	Analyze (C4)	Analyzing professional practices in software engineering

Course Title	Advanced Topics in Software Engineering
Course Code	SE-425
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	
Co-Requisite	None
Follow Up	None
Course Introduction	The Advanced Topics in Software Engineering course exposes students to the latest trends, emerging technologies, and professional practices shaping modern software development. Students will explore contemporary methodologies, frameworks, and tools, and learn to apply these technologies in real-world software engineering projects (CLO2). The course emphasizes critical analysis of industry practices, standards, and ethical considerations, enabling students to evaluate and adopt effective approaches in professional software development (CLO1, CLO3). Through case studies, hands-on exercises, and project-based learning, students gain practical experience with emerging technologies and develop the skills necessary to address evolving challenges in the software industry.
Course Description	Topics Covered (Explicit Contents): Emerging software development methodologies: DevOps, Agile, Lean, and CI/CD trends Modern programming paradigms and frameworks (functional programming, microservices, reactive frameworks) Cloud computing and serverless architectures in software engineering Application of AI/ML techniques in software projects Software quality assurance trends: automated testing, static analysis, and continuous monitoring Ethical, professional, and collaborative practices in software development Case studies of recent software projects using cutting-edge technologies Adoption and evaluation of new tools, platforms, and frameworks in professional environments Software engineering standards, best practices, and industry trends
Text Book(s)	1. Software Engineering: A Practitioner's Approach, Roger S. Pressman & Bruce R. Maxim 2. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Jez Humble & David Farley 3. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, Gene Kim, Jez Humble, Patrick Debois, John Willis 4. Designing Data-Intensive Applications, Martin Kleppmann 5. Software Engineering at Google: Lessons Learned from Programming Over Time, Titus Winters, Tom Manshreck, Hyrum Wright

Machine Learning			
CLOs	1. Explain core concepts of Machine Learning. 2. Apply machine learning algorithms to real-world datasets. 3. Analyze the performance of ML models using evaluation metrics. 4. Implement ML solutions using modern toolkits. 5. Discuss ethical considerations while applying ML to societal problems.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain core concepts of Machine Learning	PLO1, PLO2
	CLO2	Apply machine learning algorithms to real-world datasets	PLO2, PLO5
	CLO3	Analyze performance of ML models using evaluation metrics	PLO2, PLO3
	CLO4	Implement ML solutions using modern toolkits	PLO2, PLO5
	CLO5	Discuss ethical considerations while applying ML	PLO4, PLO9
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining core concepts of Machine Learning
	CLO2	Apply (C3)	Applying algorithms to real-world datasets
	CLO3	Analyze (C4)	Analyzing model performance using evaluation metrics
	CLO4	Apply (C3)	Implementing ML solutions using modern programming toolkits
	CLO5	Affective (A3)	Considering ethical implications of machine learning applications

Title	Machine Learning
Code	SE-321
Credit Hours	3 (2,1)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Introduction	Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this course is to: a) Present the basic machine learning concepts; b) Present a range of machine learning algorithms along with their strengths and weaknesses; c) Apply machine learning algorithms to solve problems of moderate complexity.
Course Description	Topics: Machine learning course begins with foundational concepts, such as understanding supervised and unsupervised learning, feature engineering, and model evaluation techniques. As the course progresses, students delve into various supervised machine learning algorithms, including linear and nonlinear regression, logistic regression, decision trees, random forest, support vector machines, and neural networks. For each of the aforementioned algorithms, students learn about their hypothesis functions, cost functions, optimization functions and regularization techniques to avoid overfitting. The topics related to unsupervised machine learning are also part of the course. The students learn clustering techniques such as k-mean's clustering, and dimensionality reduction algorithms like principal component analysis and auto-encoders. Implementation of the theoretical concepts using Python, Different libraries and frameworks such as NumPy, Pandas, Scikit Learn and TensorFlow, Libraries for supervised and unsupervised machine learning algorithms such as linear and nonlinear regression, logistic regression, decision trees, random forest, support vector machines, neural networks (forward and backward propagation), k-mean's clustering, and dimensionality reduction algorithms like principal component analysis and auto-encoders. Publicly available datasets are used for practical demonstration of the aforementioned algorithms.
Text Book(s)	1. Introduction to Machine Learning by Ethem Alpaydin
Reference Material	1. The Hundred-Page Machine Learning Book by Andriy Burkov 2. Hands On Machine Learning with Scikit Learn, Keras and TensorFlow 2e by Aurélien Géron 3. Deep Learning with Python by François Chollet 4. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Internet of Things and Its Applications			
CLOs	1. Differentiate between the layers of the IoT stack based on the TCP/IP model, demonstrating familiarity with the key technologies and protocols employed at each layer. 2. Implement IoT applications by integrating sensors and communication modules. 3. Understand performance, scalability, security, and privacy issues in IoT solutions across different application domains. 4. Build a functional IoT system by prototyping, programming, and analyzing data, ensuring the system meets specified project requirements. 5. Evaluate the role of big data, machine learning, and cloud/fog/edge computing in an IoT solution, assessing how each component contributes to the system's functionality and effectiveness.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Differentiate between the layers of the IoT stack and associated technologies	PLO1, PLO2
	CLO2	Implement IoT applications by integrating sensors and communication modules	PLO2, PLO5
	CLO3	Understand performance, scalability, security, and privacy issues in IoT solutions	PLO1, PLO4
	CLO4	Build a functional IoT system by prototyping, programming, and analyzing data	PLO2, PLO3, PLO5
	CLO5	Evaluate role of big data, ML, and cloud/fog/edge computing in IoT solutions	PLO2, PLO3, PLO6
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Analyze (C4)	Differentiating IoT stack layers and understanding technologies/protocols at each layer
	CLO2	Apply (C3)	Implementing IoT applications using sensors and communication modules
	CLO3	Understand (C2)	Understanding performance, scalability, security, and privacy aspects of IoT solutions
	CLO4	Create / Design (C5)	Building functional IoT systems by prototyping, programming, and analyzing data
	CLO5	Evaluate (C6)	Evaluating the role of big data, ML, and cloud/fog/edge computing in IoT systems

Title	Internet of Things and Its Applications
Code	SE-426
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Description	Introduction of Internet of Things (IoT), need of IoT Security, Requirement and Basic Properties, Main Challenges, Confidentiality, Integrity, Availability, Non-Repudiation, IoT Architectures (Device, Cloud, Gateway, Backend, Applications), Security Classification & Access Control, Data classification (Public and Private), Privacy issues in IoT, IoT Authentication and Authorization, IoT Data Integrity, Web Based Attacks and Implementation in IoT, Denial of Service, Sniffing, Phishing, DNS Hijacking, Pharming, Defacement etc., Cryptology Cipher –Symmetric Key Algorithms (AES and DES), Asymmetric Key Algorithm (RSA) Attacks– Dictionary and Brute Force, Lookup Tables, Reverse Lookup Tables, Rainbow Tables, Attack Surface in IoT and Threat Assessment, Embedded Device – UART, SPI, I2C, JTAG, Software and Cloud components– Firmware of the device, Web Application Dashboard, Mobile Application used to Control, Configure and Monitor the devices, Radio Communication– WIFI, BLE, Cellular, ZigBee*, ZWave*, 6LoWPAN, IoT Protocol inbuilt Security Features On Transport Layer and Application Layer, Security Management, Identity and Access Management, Key Management, Model Discussions (Smart Home, Smart Agriculture, Smart Retail Supply, Smart Healthcare, Smart Grid, Smart Cities).
Text Book(s)	1. Fei Hu, Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations, 1st Edition, ISBN-13:978-1498723183. 2. Brian Russell, Drew Van Duren, Practical Internet of Things Security, 2016

Green Software Engineering			
CLOs	1. Explain the principles of green software engineering and sustainability in the context of software design, development, and deployment. 2. Analyze energy efficiency, environmental impact, and sustainability challenges in software systems across their lifecycle. 3. Apply sustainable development practices in software design and implementation. 4. Demonstrate ethical and responsible practices in developing, maintaining, and updating software systems to support long-term social and environmental sustainability.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain principles of green software engineering and sustainability	PLO1, PLO2
	CLO2	Analyze energy efficiency, environmental impact, and sustainability in software systems	PLO2, PLO3
	CLO3	Apply sustainable development practices in software design and implementation	PLO2, PLO5
	CLO4	Demonstrate ethical and responsible practices for social and environmental sustainability	PLO4, PLO9
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining principles of green software engineering and sustainability
	CLO2	Analyze (C4)	Analyzing energy efficiency, environmental impact, and sustainability challenges
	CLO3	Apply (C3)	Applying sustainable practices in software design and implementation
	CLO4	Affective (A3)	Demonstrating ethical and responsible practices for social and environmental impact

Title	Green Software Engineering
Code	SE-427
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	None
Co-Requisite	None
Follow-up	
Course Introduction	The Green Software Engineering course introduces students to the principles and practices of sustainable software development. Students will learn how to design, develop, and deploy software systems while minimizing energy consumption and environmental impact (CLO1, CLO2). The course covers strategies for evaluating software sustainability, analyzing energy efficiency, and addressing environmental challenges across the software lifecycle. Practical sessions will focus on applying sustainable development practices in design and implementation, including efficient coding, resource optimization, and environmentally conscious deployment strategies (CLO3). Emphasis is placed on ethical and responsible software engineering practices, ensuring long-term social and environmental sustainability of software systems (CLO4). By the end of the course, students will be equipped to create software solutions that balance functionality, efficiency, and sustainability.
Course Description	Topics Covered (Explicit Technical Contents): Introduction to Green Software Engineering: principles, sustainability goals, and industry trends Environmental impact of software systems: energy consumption, carbon footprint, and lifecycle analysis Metrics and measurement for software energy efficiency Sustainable software design patterns and coding practices Resource-efficient algorithms and data structures Energy-aware software testing and profiling Green deployment strategies: cloud, edge, and serverless computing considerations Tools for monitoring and analyzing software sustainability Ethical and responsible practices in sustainable software engineering Case studies of energy-efficient and sustainable software systems Project-based learning: designing, implementing, and evaluating a sustainable software system
Text Book(s)	1. Green Software Development and Deployment, Christoph Becker, Emily B. Maier, Stefan J. Marr (Editors) 2. Sustainable Software Engineering: Concepts, Methods, and Tools, Rafael Prikladnicki, Shailesh Tiwari (Editors) 3. Software Engineering for Smart Renewable Energy Systems, Alessandro C. M. de Melo, et al.

Mobile Application Development			
CLOs	1. Explain core concepts of mobile platforms, operating systems, and mobile application architectures. 2. Design and implement mobile applications for Android and/or iOS platforms using appropriate tools and frameworks. 3. Apply UI/UX principles and mobile-specific design patterns to develop intuitive and responsive user interfaces. 4. Test, debug, and deploy mobile applications on actual devices, ensuring performance and reliability. 5. Evaluate security, privacy, and performance considerations in mobile applications, incorporating best practices.		
CLO–PLO Mapping	CLO	Exact Statement (Short)	PLO Mapping
	CLO1	Explain core concepts of mobile platforms, OS, and app architectures	PLO1, PLO2
	CLO2	Design and implement mobile applications using appropriate tools and frameworks	PLO2, PLO3, PLO5
	CLO3	Apply UI/UX principles and mobile design patterns to develop intuitive interfaces	PLO2, PLO5
	CLO4	Test, debug, and deploy mobile applications ensuring performance and reliability	PLO2, PLO3, PLO5
	CLO5	Evaluate security, privacy, and performance considerations in mobile applications	PLO2, PLO3, PLO4
Bloom's Taxonomy Mapping	CLO	Bloom Level	Reason
	CLO1	Understand (C2)	Explaining mobile platforms, operating systems, and application architectures
	CLO2	Create / Design (C5)	Designing and implementing mobile applications using development tools and frameworks
	CLO3	Apply (C3)	Applying UI/UX principles and design patterns to build user-friendly interfaces
	CLO4	Apply → Analyze (C3–C4)	Testing, debugging, and deploying apps while analyzing performance and reliability
	CLO5	Evaluate (C5)	Evaluating security, privacy, and performance considerations for mobile applications

Course Title	Mobile Application Development
Course Code	SE-325
Credit Hours	3 (3,0)
Category	Software Engineering Specialization
Prerequisite	CC-211 Object Oriented Programming
Co-Requisite	None
Follow-up	None
Course Description	Introduction: Mobile Computing Domain, Comparison of different Mobile Platforms, Revision of Java. Android Platform: Dalvik Virtual Machine, Android Constructs (Activity, Intent, Content Provider, Service, Broadcast Receiver), Activity Lifecycle, Project Structure, Manifest File, Emulators (AVD), Debugging (DDMS), R.java, Inter Activity Communication (Intent), Event Handlers, Layout XML, @ Sign, Layouts, Adapters, Dynamic Lists, Holder Pattern, Menus and dialogs, Menu, Menu Group, Menu Item, Icon Menu, Sub Menu, Context Menu, Sub Menu, Dynamic Menu, Using XML Files for Menus, Services, Intents. Data Storage: Key Value Sets, Files, Intro to SQLite, Web Service Integration, JSON, HTTPClient, Graphics, Widgets & Notifications, Multimedia and telephony API, Android Native Development Kit (NDK).
Text Book(s)	1. Mark L. Murphy, Beginning Android 2, Apress, ISBN 978-1-4302-2629-1 2. Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura Programming Android, 2nd Edition, O Reilly Press, ISBN: 978-1-449-31664-8 3. Syed Hashmi, Satya Komatineni, Dave Maclean, Pro Android 2, Apress, ISBN 978-1-4302-2659-8
Reference Material	1. http://developer.android.com/develop/index.htm 2. http://developer.android.com/guide/components/services.html 3. http://developer.android.com/guide/components/intents-filters.html 4. http://developer.android.com/guide/topics/data/data-storage.html 5. http://developer.android.com/guide/topics/graphics/index.html 6. http://developer.android.com/guide/topics/sensors/index.html

General Education Courses: 34 Credits (14 Courses)

This section outlines the General Education (GE) courses designed to provide students with a broad foundation of knowledge and skills beyond their core discipline. The courses aim to develop critical thinking, communication, quantitative reasoning, and ethical awareness, enabling students to become well-rounded professionals. Each GE course is presented with its Course Learning Outcomes (CLOs), mapping to relevant Program Learning Outcomes (PLOs), and alignment with Bloom's Taxonomy, ensuring that students acquire both conceptual understanding and practical competencies applicable across diverse contexts.

Course Title	Application of Information & Communication Technologies		
Course Code	GE-160		
Credit Hours	3 (2,1)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow Up	None		
Course Introduction	The course is designed to provide students with an exploration of the practical applications of Information and Communication Technologies (ICT) and software tools in various domains. Students will gain hands-on experience with a range of software applications, learning how to leverage ICT to solve daily life problems, enhance productivity and innovate in different fields. Through individual and interactive exercises and discussions, students will develop proficiency in utilizing various software related to ICT.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Explain the fundamental concepts, components, and scope of ICT.	C2 (Explain)	1,2
	CLO2: Identify uses of various ICT platforms and tools for different purposes.	C4 (Identify)	1,2,3
	CLO3: Apply ICT platform and tools for different purposes to address basic needs in different domains of daily, academic, and professional life.	C3 (Apply)	3,4,5
	CLO4: Understand ethical and legal considerations in use of ICT platforms and tools.	C2 (Understand)	1,2,3
Course Description	Introduction to ICT Components of ICT (basics of hardware, software, ICT platforms, networks, local and cloud data storage, etc.), Scope of ICT (use of ICT in education, business, governance, healthcare, digital media and entertainment, etc.), Emerging technologies and future trends. Basic ICT Productivity Tools: Effective use of popular search engines to explore WWW, Formal communication tools and etiquettes (Gmail, Microsoft Outlook, etc.) Microsoft Office Suites (Word, Excel, PowerPoint), Google Workspace (Google Docs, Sheets, Slides), Google Drive, Dropbox (cloud storage and file sharing), Google Drive (Cloud storage with Google Docs integration) and Microsoft OneDrive (Cloud storage with Microsoft integration), Evernote (Note-taking and organization applications) and OneNote (Microsoft's digital notebook for capturing and organizing ideas), Video conferencing (Google Meet, Microsoft Teams, Zoom, etc.), social media applications (LinkedIn, Facebook, Instagram, etc.)		

	ICT in Education: Working with learning management systems (Moodle, Canvas, Google Classrooms, etc.), Sources of online education courses (Coursera, edX, Udemy, Khan Academy, etc.), Interactive multimedia and virtual classrooms ICT in Health and Well-being: Health and fitness tracking devices and applications (Google Fit, Samsung Health, Apple Health, Xiaomi Mi Band, Runkeeper, etc.), Telemedicine and online health consultations (OLADOC, Sehat Kahani, Mahram, etc.) ICT in Personal Finance and Shopping: Online banking and financial management tools (JazzCash, Easypaisa, Zong PayMax, 1Link and MNET, Keenu Wallet, etc.), E-commerce platforms. Digital Citizenship and Online Etiquette: Intellectual property and copyright issues, Ensuring originality in content creation by avoiding plagiarism and unauthorized use of information sources, Content accuracy and integrity (ensuring that the content share through ICT platforms is free from misinformation, fake news, and manipulation).
Practical Requirements	1. Guided tutorials and exercises to ensure that students are proficient in commonly used software applications such as word processing software (e.g., Microsoft Word), presentation software, (e.g., Microsoft PowerPoint), spreadsheet software (e.g., Microsoft Excel) among such other tools. Students may be assigned practical tasks that require them to create documents, presentations, and spreadsheets etc. 2. Assigning of tasks that involve creating, managing, and organizing files and folders on both local and cloud storage systems. Students will practice file naming conventions, creating directories, and using cloud storage solutions (e.g., Google Drive, OneDrive). 3. The use of online learning management systems (LMS) where students can access course materials, submit assignments, participate in discussion forums, and take quizzes or tests. This will provide students with the practical experience with online platforms commonly used in education and the workplace.
Suggested Instructional/ Reading Material	1. “Discovering Computers” by Vermaat, Shaffer, and Freund. 2. Deborah Morley and Charles S. Parker, Understanding Computers: Today and Tomorrow, 16th edition, Cengage Learning, 2016, ISBN-13: 978-1337251853 3. “Computing Essentials” by Morley and Parker. 4. “GO! With Microsoft Office” Series by Gaskin, Vargas, and McLellan. 5. “Exploring Microsoft Office” Series by Grauer and Poatsy. 6. “Technology in Action” by Evans, Martin and Poatsy. 7. Livesley, Robert Kenneth. An introduction to automatic digital computers. Cambridge University Press, 2017. 8. Joan Lambert, Curtis Frye, Microsoft Office 2019 Step by Step, First Edition. ISBN: 978-1-50-930597-1.

Title	Functional English		
Code	GE-162		
Credit Hours	3 (3,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	GE-165 Expository Writing		
Course Introduction	This course is designed to equip students with essential language skills for effective communication in diverse real-world scenarios. It focuses on developing proficiency in English language usage: word choices, grammar and sentence structure. In addition, the course will enable students to grasp nuanced messages and tailor their communication effectively through application of comprehension and analytical skills in listening and reading. Moreover, the course encompasses a range of practical communication aspects including professional writing, public speaking, and everyday conversation, ensuring that students are equipped for both academic and professional spheres. An integral part of the course is fostering a deeper understanding of the impact of language on diverse audiences. Students will learn to communicate inclusively and display a strong commitment to cultural awareness in their language use. Additionally, the course will enable them to navigate the globalized world with ease and efficacy, making a positive impact in their functional interactions.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Apply enhanced English communication skills through effective use of word choices, grammar and sentence structure.	C4 (Apply)	1,6,7,10
	CL2: Comprehend a variety of literary / non-literary written and spoken texts in English.	C2 (Understand)	1,6,7,10
	CLO3: Effectively express information, ideas and opinions in written and spoken English.	C2 (Explain)	1,6,7,10
	CLO4: Recognize inter-cultural variations in the use of English language and to effectively adapt their communication style and content based on diverse cultural and social contexts.	C4 (Identify)	1,6,7,10
Course Description	Foundations of Functional English: Vocabulary building (contextual usage, synonyms, antonyms and idiomatic expressions), Communicative grammar (subject-verb-agreement, verb tenses, fragments, run-ons, modifiers, articles, word classes, etc.), Word formation (affixation, compounding, clipping, back formation, etc.), Sentence structure (simple, compound, complex and compound-complex), Sound production and pronunciation. Comprehension and Analysis: Understanding purpose, audience and context, Contextual interpretation (tones, biases, stereotypes, assumptions, inferences, etc.), Reading strategies (skimming, scanning, SQ4R, critical reading, etc.), Active listening (overcoming listening barriers, focused listening, etc.). Effective Communication: Principles of communication (clarity, coherence, conciseness, courteousness, correctness, etc.), Structuring documents (introduction, body, conclusion and formatting), Inclusivity in communication (gender-neutral language, stereotypes, cross-cultural communication, etc.), Public speaking (overcoming stage fright, voice modulation and body language), Presentation skills (organization content, visual aids and engaging the audience), Informal communication (small talk, networking and conversational skills), Professional writing (business e-mails, memos, reports, formal letters, etc.)		

Practical Requirements	As part of the overall learning requirements, students will also be exposed to relevant simulations, role-plays and real-life scenarios and will be required to apply skills acquired throughout the course in the form of a final project.
Suggested Instructional / Reading Material	1. “Understanding and Using English Grammar” by Betty Schramper Azar. 2. “English Grammar in Use” by Raymond Murphy. 3. “The Blue Book of Grammar and Punctuation” by Jane Straus. 4. “English for Specific Purposes: A Learning-Centered Approach” by Tom Hutchinson and Alan Waters. 5. “Cambridge English for Job-hunting” by Colm Downes. 6. “Practical English Usage” by Michael Swan. 7. “Reading Literature and Writing Argument” by Missy James and Alan P. Merickel. 8. “Improving Reading: Strategies, Resources, and Common Core Connections” by Jerry Johns and Susan Lenski. 9. “Comprehension: A Paradigm for Cognition” by Walter Kintsch. 10. “Communication Skills for Business Professionals” by J.P. Verma and Meenakshi Raman.

Title	Expository Writing		
Code	GE-165		
Credit Hours	3 (3,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	GE-261: Technical and Business Writing		
Course Introduction	Expository Writing is a sequential undergraduate course aimed at refining writing skills in various contexts. Building upon the foundation of the pre-requisite course, Functional English, this course will enhance students' abilities of producing clear, concise and coherent written texts in English. The course will also enable students to dissect intricate ideas, to amalgamate information and to express their views and opinions through well-organized essays. The students will further be able to refine their analytical skills to substantiate their viewpoints using credible sources while adhering to established ethical writing norms. Additionally, the course will highlight the significance of critical thinking enabling students to produce original and engaging written texts.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Understand the essentials of the writing process integrating pre-writing, drafting, editing and proof reading to produce well-structured essays.	C2 (Understand)	1,6,7,10
	CLO2: Demonstrate mastery of diverse expository types to address different purposes and audiences.	C3 (Apply)	1,6,7,10
	CLO3: Uphold ethical practices to maintain originality in expository writing.	C3 (Demonstrate)	1,6,7,9

Course Description	Introduction to Expository Writing: Understanding expository writing (definition, types, purpose and applications), Characteristics of effective expository writing (clarity, coherence and organization), Introduction to paragraph writing. The Writing Process: Pre-writing techniques (brainstorming, free-writing, mind-mapping, listing, questioning and outlining etc.), Drafting (three stage process of drafting techniques), Revising and editing (ensuring correct grammar, clarity, coherence, conciseness etc.), Proof reading (fine-tuning of the draft), Peer review and feedback (providing and receiving critique). Essay Organization and Structure: Introduction and hook (engaging readers and introducing the topic), Thesis statement (crafting a clear and focused central idea), Body Paragraphs (topic sentences, supporting evidence and transitional devices), Conclusion (types of concluding paragraphs and leaving an impact), Ensuring cohesion and coherence (creating seamless connections between paragraphs). Different Types of Expository Writing: Description, Illustration, Classification, Cause and effect (exploring causal relationships and outcomes), Process analysis (explaining step-by-step procedures), Comparative analysis (analyzing similarities and differences). Writing for Specific Purposes and Audiences: Different types of purposes (to inform, to analyze, to persuade, to entertain etc.), Writing for academic audiences (formality, objectivity, and academic conventions), Writing for public audiences (engaging, informative and persuasive language), Different tones and styles for specific purposes and audiences. Ethical Considerations: Ensuring original writing (finding credible sources, evaluating information etc.), Proper citation and referencing (APA, MLA, or other citation styles), Integrating quotes and evidences (quoting, paraphrasing, and summarizing), Avoiding plagiarism (ethical considerations and best practices)
Practical Requirements	As part of the overall learning requirements, students will be required to build a writing portfolio having a variety of expository texts and present the same at the end of the course showcasing proficiency in expository writing.
Suggested Instructional / Reading Material	1. "The St. Martin's Guide to Writing" by Rise B. Axelrod and Charles R. Cooper. 2. "They Say / I Say: The Moves That Matter in Academic Writing" by Gerald Graff and Cathy Birkenstein. 3. "Writing Analytically" by David Rosenwasser and Jill Stephen. 4. "Style: Lessons in Clarity and Grace" by Joseph M. Williams and Joseph Bizup. 5. "The Elements of Style" by William Strunk Jr. and E.B. White. 6. "Good Reasons with Contemporary Arguments" by Lester Faigley and Jack Selzer. 7. "Writing to Learn; How to Write – and Think – Clearly About Any Subject at All" by William Zinsser. 8. "The Norton Field Guide to Writing" by Richard Bullock, Maureen Daly Goggin, and Francine Weinberg. 9. "The Art of Styling Sentences" by Ann Longknife and K.D. Sullivan. 1. "Writing Today" by Richard Johnson-Sheehan and Charles Paine.

Course Title	Discrete Structures		
Course Code	GE-161		
Credit Hours	3 (3,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Understand the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs and Trees etc.	C2 (Understand)	1,3
	CLO2: 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)	1,3
	CLO3: Apply discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.	C3 (Apply)	1,2,3,4
	CLO4: 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)	1,2,3,4
Course Description	Mathematical Reasoning: Propositional and predicate logic. Propositional Logic: Logical operators, translations between symbolic expressions and formal English expression, logical equivalences. Predicate Logic: Quantifiers, Nested quantification, equivalences, translations between symbolic forms and formal English. Rules of Inference: Proof methods and strategies, Direct proof, Proof by contraposition, proof by induction, proof by implication, Existence proof, Uniqueness proofs, trivial proofs, vacuous proofs. Sets: Notations, set operations, Venn diagrams, countable and uncountable sets, relations, equivalence relations and partitions, partial orderings, recurrence relations, functions, mappings. Functions: Injective, surjective, bijective, special types of functions, function composition, inverse functions, recursive functions, compositions, number theory, sequences, series, counting, inclusion and exclusion principle, pigeonhole principle, permutations and combinations. Integers and Divisibility: Division theorem, modular arithmetic, LCM, GCD, Euclidean and Extended Euclidean method, finding solutions to congruence. Primes: Fundamental theorem of arithmetic, characterizations of primes, Mersenne primes. Induction: Weak induction, strong induction. Recursion and Recurrences: Formulation of recurrences, closed formulas, Counting: product rule, sum rule, principle of inclusion-exclusion, combinations and permutations, binomial coefficients, Pascal's identity and Pascal's triangle, binomial theorem, pigeonhole principle. Relations: Reflexive, symmetric, transitive, antisymmetric, equivalence relations and equivalence classes, partial orders. Graph Theory: Terminologies, elements of graph theory, planar graphs, graph coloring, Euler graph, Hamiltonian path, rooted trees, traversals, handshaking lemma and corollary, special families of graphs, isomorphism, planarity, Eulerian and Hamiltonian graphs, trees.		
Text Book(s)	1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, 7 th Edition, McGraw Higher-Ed, 2011, ISBN: 0073383090.		
Reference Material	1. Susanna S. Epp, Discrete Mathematics with Applications, 4th Edition. 2. Richard Johnson Baugh, Discrete Mathematics, 7th Edition.		

	3. Kolman, Busby & Ross, Discrete Mathematical Structures, 4th Edition. 4. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction, 5th Edition. 5. Winifred Grassman, Logic and Discrete Mathematics: A Computer Science Perspective, 1st Edition.
--	--

Title	Islamic Studies		
Code	GE-264		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	This course is designed to provide students with a comprehensive overview of the fundamental aspects of Islam, its benefits, practices, history and influence on society. It will further familiarize the students with a solid foundation in understanding Islam from an academic and cultural perspective. Through this course, students will have an enhanced understanding of Islam's multifaceted dimensions which will enable them to navigate complex discussions about Islam's historical and contemporary role, fostering empathy, respect, and informed dialogue.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Demonstrate enhanced knowledge of Islamic foundational beliefs, practices, historical development, fundamental sources of Shiarah, spiritual values and ethical principles.	C1 (Knowledge)	1,7,9,10
	CLO2: Describe basic sources of Islamic law and their application in daily life.	C2 (Understand)	1,7,9,10
	CLO3: Identify and discuss contemporary issues being faced by the Muslim world including social challenges, gender roles and interfaith interactions.	C4 (Analyze)	1,6,7,9,10

[illegible]

6. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: أَتَكُونُونَ مَا الْبُطْلُ قَالَوا: أَلَيْسَ فِيمَا مِنْ لَدُنْهُمْ لَهْمٌ لَا مَتَاعَ فِيهِ قَالَ: بَلَى الْبُطْلُ مِنْ لَدُنِّي وَمِنْ يَوْمِ الْقِيَامَةِ يَصْلَوْنَ صِغِيرًا وَكُنَا قَوْمًا عَلَى شَرِّ شَيْءٍ هَذَا وَقَدْ هَذَا أَكَلَّ مَالَهُ اسْمُهُ هَذَا وَهَذَا اسْمُهُ هَذَا الْفَيْعُ عَلَى هَذَا مِنْ حَسْبِ مَا تَمَّ قَوْلِي فَيَمُوتُ حَسْبُ مَا تَمَّ قَوْلِي نِي فَهِيَ مَا عَلَيْنَا مِنْ غَطِّ آيَاتِهِمْ فَطَرَحَتْ عَلَيْهِمْ طَرَحِي النَّارِ. (رواه مسلم، كتاب البر: 6579)
7. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
8. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
9. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
10. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
11. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
12. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
13. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
14. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
15. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)

16. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)
17. عَنْ عَبْدِ اللَّهِ بْنِ مَرْثَدَةَ عَنْ رَسُولِ اللَّهِ ﷺ قَالَ: كَانَ رَسُولُ اللَّهِ ﷺ يَقُولُ لِللَّهِمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكَرْبِ وَمِنْ عِلْمٍ لَا يَنْفَعُ وَمِنْ دَعْوَةٍ لَا تُنْفَعُ وَمِنْ قَلْبٍ لَا يَضَعُ وَمِنْ نَفْسٍ لَا تَضَعُ. (رواه مسلم و أحمد سنن ابن ماجه: 250)

3- ایمانیات و عبادات: تعارف اور مختصر۔

- 4- سیرت نبوی صلی اللہ علیہ وآلہ وسلم
- 1- رسول اللہ صلی اللہ علیہ وآلہ وسلم کی حیات طیبہ (حالات زندگی قبل از بعثت، اعلان نبوت، ہجرت مدینہ، غزوات، صلح حدیبیہ، خطبہ حجۃ الوداع)
- 2- رسول اللہ ﷺ کی مختلف حیثیات: (i) بحیثیت معلم (ii) بحیثیت پیغمبر امن

	5۔ خلافتِ راشدہ ۱۔ خلفائے راشدین (حضرت ابو بکر صدیقؓ، حضرت عمر فاروقؓ، حضرت عثمان غنیؓ، حضرت علی المرتضیٰؓ) کے شخصی احوال ۲۔ عہد خلافتِ راشدہ کے اہم خصائص 6۔ فقہِ اسلامی 1۔ فقہِ اسلامی کے مکتبہ (مذہب) آج، سنی، اہل حنفیہ، اہل مالک، اہل شافعی، اہل حنبلیہ 2۔ ائمہ اربعہ کا تعارف (امام ابو حنیفہؒ، امام مالکؒ، امام شافعیؒ، امام احمد بن حنبلؒ) 7۔ اسلام کی سماجی تعلیمات 1۔ خاندان کا تعارف و اہمیت 2۔ اسلام میں عورت کا مقام 3۔ اسلام کا قصور حقوق العباد 4۔ اسلام کی اخلاقی تعلیمات 8۔ اسلام اور جدید دنیا 1۔ عالمگیری اور اس کے مسائل و تقاضے 2۔ اسلام و نوپیا 3۔ کثیرتہ معاشرہ Pluralistic Society اور اسلامی تعلیمات
Suggested Instructional/ Reading Material	1. "Introduction to Islam" by Dr. Muhammad Hamidullah 2. "Principles of Islamic Jurisprudence" by Dr. Ahmad Hasan 3. "Muslim Jurisprudence and the Quranic Law of Crimes" by Mir Waliullah 4. "Sirat-un-Nabi" by Shibli Nomani and Sulaiman Nadvi

Title	Ideology and Constitution of Pakistan		
Code	GE-261		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	This course is designed to provide students with a fundamental exploration of the ideology and the constitution of Pakistan. The course focuses on the underlying principles, beliefs, and aspirations that have been instrumental in shaping the creation and development of Pakistan as a sovereign state. Moreover, the course will enable students to understand the core provisions of the Constitution of the Islamic Republic of Pakistan concerning the fundamental rights and responsibilities of Pakistan citizens to enable them function in a socially responsible manner.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Demonstrate enhanced knowledge of the basis of the ideology of Pakistan with special reference to the contributions of the founding father of Pakistan.	C3 (Demonstrate)	1,10
	CLO2: Demonstrate fundamental knowledge about the Constitution of Pakistan 1973 and its evolution with special reference to state structure.	C3 (Demonstrate)	1,10
	CLO3: Explain about the guiding principles on rights and responsibilities of Pakistan citizens as enshrined in the Constitution of Pakistan 1973.	C2 (Understand)	1,3,10
Course Description	1. Introduction to the Ideology of Pakistan: Definition and significance of ideology, Historical contest of the creation of Pakistan (with emphasis on socio-political religious and cultural dynamics of British India between 1857 till 1947), Contributions of founding fathers of Pakistan of Pakistan in the freedom movement including but not limited to Allama Muhammad Iqbal, Muhammad Ali Jinnah., etc. Contributions of women and students in the freedom movement for separate homeland for Muslims of British India. Two-Nation Theory: Evolution of the Two-Nation Theory (Urdu-Hindi controversy, Partition of Bengal, Simla Deputation 1906, Allama Iqbal's Presidential Address 1930, Congress Ministries 1937 Lahore Resolution 1940). Introduction to the Constitution of Pakistan: Definition and importance of a constitution, Ideological factors that shaped the Constitution(s) of Pakistan (Objectives Resolution 1949). Constitution and State Structure: Structure of Government (executive, legislature, and judiciary), Distribution of powers between federal and provincial governments, 18th Amendment and its impact on federalism. Fundamental Right, Principles of Policy and Responsibilities: Overview of fundamental rights guaranteed to citizens by the Constitution of Pakistan 1973 (Articles 8-28), Overview of Principles of Policy (Articles 29-40), Responsibilities of the Pakistan citizens (Article 5). Constitutional Amendments: Procedures for amending the Constitution, Notable Constitutional amendments and their implications		

Suggested Instructional / Reading Material	1. "The Idea of Pakistan" by Stephen P. Cohen. 2. "Ideology of Pakistan" by Javed Iqbal. 3. "The Struggle for Pakistan" by I.H. Qureshi. 4. "Pakistan the Formative Phase" by Khalid Bin Sayeed. 5. "Pakistan: Political Roots and Development" by Safdar Mahmood. 6. "Ideology of Pakistan" by Sharif-ul-Mujahid. 7. "The Struggle for Pakistan: A Muslim Homeland and Global Politics" by Ayesha Jala. 8. "Jinnah, Pakistan and Islamic Identity: The Search for Saladin" by Akbar S. Ahmed. 9. "The Making of Pakistan: A Study in Nationalism" by K.K. Aziz. 10. "Pakistan: A New History" by Lan Talbot. 11. "Pakistan in the Twentieth Century: A Political History" by Lawrence Ziring. 12. "The Constitution of Pakistan 1973". Original. 13. "Constitutional and Political Development of Pakistan" by Hamid Khan. 14. "The Parliament of Pakistan" by Mahboob Hussain. 15. "Constitutional Development in Pakistan" by G.W. Choudhury. 16. "Constitution-Making in Pakistan: The Dynamics of Political Order" by G.W. Choudhury.
---	---

Title	Introduction to Management		
Code	GE-192		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	This course is designed to provide a comprehensive overview of organizational management. It covers the diverse roles of managers, the crucial interplay between an organization's mission, its goals, and its detailed objectives, and the impact of both internal and external environmental factors on organizations and how they strategize in response. Students will gain empirical insights into organizational processes, behaviors, and their foundational theories. Emphasis will be placed on honing critical thinking, particularly in addressing ethical dilemmas, global perspectives, and diversity within management functions. Finally, this course sheds light on the intricacies of organizational design and structural challenges, providing a holistic grasp of management dynamics.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Understand and apply management principles and concepts as they apply to business situations.	C2 (Understand)	1,3,6,7,9,10
	CLO2: Understand the role of management in making business decisions.	C2 (Understand)	1,3,6,7,9,10
	CLO3: Understand efficiently and effectively working in any kind of organization.	C2 (Understand)	1,3,6,7,9,10
Course Description	Managing and the Managers Job: The management process, kinds of managers, basic managerial roles and skills, and the nature of managerial work. The Environment and Culture of Management: The external and internal environment, the organization's cultures, organization-environment relationship. Planning and Decision Making: Decision making and planning process, organizational goals and levels of planning, various levels of strategies, rational perspectives on decision making, behavioral aspects of decision making, group and team decision making in organizations. The Organizing Process: Designing Jobs, grouping jobs, establishing reporting relationships, distributing authority, coordinating activities. Managing Change and Innovation: Forces for change, steps in the change process, understanding and overcoming resistance to change, the innovation process. Leadership and Influence Process: Different approaches to leadership, political behavior in organizations. The Controlling Process: The purpose of control, steps in the control process; operational, structural, and strategic control, managing total quality and productivity. The Ethical and Social Environment: Individual ethics in organization, emerging ethical issues, social responsibility and organizations.		
Text Book(s)	1. Ricky W. Griffin, Management, 12 th Edition, Cengage Learning, 2017, ISBN: 1305501292.		
Reference Material	1. Stephen P Robbins, Mary Coulter, Management, 14 th Edition, Pearson, 2017, ISBN: 0134527607.		

Course Title	Applied Physics		
Course Code	GE-262		
Credit Hours	3 (2,1)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow Up	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.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: To understand the fundamental concepts of Physics.	C2 (Understand)	1,2
	CLO2: To understand about charges and their interactions.	C2 (Understand)	1,2
	CLO3: To develop strong concepts of numerical techniques related to vectors and electrostatics and magnetism.	C2 (Understand)	1,2
	CLO4: To develop the relation between electricity and magnetism.	C4 (Analyze)	1,2,3
Course Description	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.		
Text Book(s)	1. D. Halliday, R. Resnick, Kenneth S. Krane, Physics Vol. 2, 5th Ed., John Wiley, 2001, ISBN: 978-0471401940.		
Reference Material	1. Hugh D. Young, Roger A. Freedman, A. Lewis, Sears, University Physics, 11th Ed., Benjamin-Cummings Pub. Co., 2004, ISBN: 978-0805391794. 2. D. Halliday, R. Resnick, J. Walker, Fundamentals of Physics, 6th Ed., Wiley, 2010, ISBN: 978-0470469118. 3. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker 4. Narciso Garcia, Arthur Damask, Steven Schwarz., "Physics for Computer Science Students", Springer Verlag, 1998.		

Course Title	Professional Practices		
Course Code	GE-263		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	A computing graduate as a professional has some responsibilities with respect to society. This course develops student understanding of historical, social, economic, ethical, and professional issues related to the discipline of computing. It identifies key sources for information and opinions about professionalism and ethics. Students analyze, evaluate, and assess ethical and professional computing case studies.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Trace the historical evolution of the computing profession and its impact on society.	C1 (Knowledge)	1,8
	CLO2: Describe the interplay between computing technologies and societal shifts, acknowledging both positive and negative implications.	C2 (Describe)	1,8,9,10
	CLO3: Recognize and explain the core ethical principles that guide the computing profession.	C4 (Identify)	1,8,9,10
	CLO4: Explain the responsibilities of computing professionals in their interactions with society and individuals.	C2 (Explain)	1,8,9,10
	CLO5: Analyze and critically evaluate real-world case studies in computing, assessing them from both ethical and professional viewpoints.	C4 (Analyze)	1,8,9,10
Course Description	Historical, social, and economic context of computing (software engineering, computer science, and information technology); definitions of computing (software engineering, computer science, and information technology) subject areas and professional activities; professional societies; professional ethics; professional competency and life-long learning; uses, misuses, and risks of software; information security and privacy; business practices and the economics of software; intellectual property and software law (cyber law); social responsibilities; software-related contracts; software house organization. Intellectual property rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse, and the Criminal Law, Regulation, and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, and the ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.		
Text Book(s)	1. Michael J. Quinn, Ethics for the Information Age, 7th Edition, Pearson Education, 2017, ISBN: 978-0134296548 2. Sara Baase, A Gift of Fire: Social, Legal, and Ethical Issues in Computing, 5th Edition, Pearson, 2018, ISBN: 978-0132492676		
Reference Material	1. J. Kizza, Ethical and Social Issues in the Information Age, 6th Edition, Springer, 2017, ISBN: 978-3319707112		

- | | |
|--|--|
| | <ol style="list-style-type: none">2. "Professional Issues in Software Engineering" by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, 3rd Edition, CRC Press, 2000. ISBN-10: 07484095133. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition, 2009. ISBN-10: 0131112414 |
|--|--|

Title	Civics and Community Management		
Code	GE-260		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	This course is designed to provide students with fundamental knowledge about civics, citizenship, and community engagement. Students will learn about the essentials of civil society, government, civic responsibilities, inclusivity, and effective ways to participate in shaping the society which will help them apply theoretical knowledge to the real-world situations to make a positive impact on their communities.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Demonstrate fundamental understanding of civics, government, citizenship and civil society.	C3 (Demonstrate)	1
	CLO2: Understand the concept of community and recognize the significance of community engagement for individuals and groups.	C2 (Understand)	1,3,6,7,9,10
	CLO3: Recognize the importance of diversity and inclusivity for societal harmony and peaceful co- existence.	C4 (Identify)	1,6,7,9,10
Course Description	Introduction to Civics and Citizenship: Definition of civics, citizenship, and civic engagement, Historical evolution of civic participation, Types of citizenship: active, participatory, digital, etc., The relationship between democracy and citizenship. Civics and Citizenship: Concepts of civics, citizenship, and civic engagement, Foundations of modern society and citizenship, Types of citizenship: active, participatory, digital, etc., State, Government and Civil Society: Structure and functions of government in Pakistan, the relationship between democracy and civil society, right to vote and importance of political participation and representation. Rights and Responsibilities: Overview of fundamental rights and liberties of citizens under Constitution of Pakistan 1973, Civic responsibilities and duties, Ethical considerations in civic engagement (accountability, non-violence, peaceful dialogue, civility, etc.) Community Engagement: Concept, nature and characteristics of community, Community development and social cohesion, Approaches to effective community engagement, Case studies of successful community driven initiatives. Advocacy and Activism: Public discourse and public opinion, Role of advocacy in addressing social issues, Social action movements. Digital Citizenship and Technology: The use of digital platforms for civic engagement, Cyber ethics and responsible use of social media, Digital divides and disparities (access, usage, socioeconomic, geographic, etc.) and their impacts on citizenship. Diversity, Inclusion and Social Justice: Understanding diversity in society (ethnic, cultural, economic, political etc.), Youth, women and minorities' engagement in social development, Addressing social inequalities and injustices in Pakistan, Promoting inclusive citizenship and equal rights for societal harmony and peaceful co-existence.		

Text Book(s)	1. "Civics Today: Citizenship, Economics, & You" by McGraw-Hill Education 2. "Citizenship in Diverse Societies" by Will Kymlicka and Wayne Norman. 3. "Digital Citizenship in Action: Empowering Students to Engage in Online Communities" by Kristen Mattson. 4. "Globalization and Citizenship: In the Pursuit of a Cosmopolitan Education" by Graham Pike and David Selby.
Reference Material	1. "Community Engagement: Principles, Strategies, and Practices" by Becky J. Feldpausch and Susan M. Omilian. 2. "Creating Social Change: A Blueprint for a Better World" by Matthew Clarke and Marie-Monique Steckel.

Course Title	Entrepreneurship		
Code	GE-265		
Credit Hours	2 (2,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Introduction	This course is designed to promote entrepreneurial spirit and outlook among students, encouraging them to think critically, identify opportunities, and transform their ideas into successful ventures. It aims at imparting them with the requisite knowledge; skills and abilities, enabling them seize the identified opportunities for initiating of business (including requirements for registration and incorporation with regulators such as SECP and others), market research, opportunity identification, business planning, financial literacy for managing finances and securing funding, marketing and sales, team building and innovation, overall, the course is geared towards personal growth and professional development for pursuing innovative ideas, availing opportunities and initiating start-ups.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Knowledge of fundamental entrepreneurial concepts, skills and process	C1 (Knowledge)	1
	CLO2: Understanding of different personal, social and financial aspects associated with entrepreneurial activities	C2 (Understand)	1,3,10
	CLO3: Basic understanding of regulatory requirements to set up an enterprise in Pakistan, with special emphasis on exports	C2 (Understand)	1,3,10
	CLO4: Ability to apply knowledge, skills and abilities acquired in the course to develop a feasible business plan for implementation	C5 (Create)	4,5,6,10
Course Description	Introduction: Definition and concept of entrepreneurship, Why to become an entrepreneur? Entrepreneurial process, Role of entrepreneurship in economic development. Entrepreneurial Skills: Characteristics and qualities of successful entrepreneurs (including stories of successes and failures), Areas of essential entrepreneurial skills and abilities such as creative and critical thinking innovation and risk taking. Opportunity Recognition and Idea Generation: Opportunity identification, evaluation and exploitation, Innovative ideas generation techniques for entrepreneurial ventures. Marketing and sales: Four P's of Marketing, Developing a marketing strategy, Branding. Financial Literacy: Basic concepts of income, savings and investments, Basic concepts of assets, liabilities and equity, Basic concepts of revenue and expenses, Overview of cash-flows, Overview of banking products including Islamic modes of financing, Sources of funding for startups (angel financing, debt financing, equity financing etc.) Team Building for Startups: Characteristics and features of effective teams, Team building and effective leadership for startups. Regulatory Requirements to Establish Enterprises in Pakistan: Types of enterprises (e.g., sole proprietorship; partnership; private limited companies etc.), Intellectual property rights and protection, Regulatory requirements to register an enterprise in Pakistan, with special emphasis on exports firms, Taxation and financial reporting obligation.		
Practical Requirements	As part of the overall learning requirements, students shall be tasked with creating and presenting a comprehensive business plan at the end of the course for a hypothetical or real		

	business idea. This practical exercise shall allow them to apply the knowledge, skills and abilities acquired in the course to develop a feasible business plan and where possible explore the possibility of implementing the plan with support and assistance from established business-persons and entrepreneurs.
Suggested Instructional/ Reading Material	1. B. R. Barringer, and R. D. Ireland, Entrepreneurship: Successfully Launching New Ventures, 6th Edition, Prentice Hall, 2019, ISBN: 978-0134729534. 2. "Entrepreneurship: Theory, Process, and Practice" by Donald F. Kuratko. 3. "New Venture Creation: Entrepreneurship for the 21st Century" by Jeffry A. Timmons, Stephen Spinelli Jr., and Rob Adams. 4. "Entrepreneurship: A Real-World Approach" by Rhonda Abrans. 5. "The Lean Startup: How Today's Entrepreneurs use Continuous Innovation to Create Radically Successful Businesses" by Eric Ries. 6. "Effectual Entrepreneurship" by Stuart Read, Saras Sarasvathy, Nick Dew, Robert Wiltbank, and Anne-Valcrie Ohlsson.

Course Title	Probability and Statistics		
Course Code	GE-164		
Credit Hours	3 (3,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Know the concept and applications of probability and statistics.	C1 (Know)	1
	CLO2: Describe expectation and distributions	C2 (Describe)	1
	CLO3: Solve single sample and one- and two- sample estimation.	C3 (Apply)	1,3
	CLO4: Use regression techniques.	C3 (Apply)	1,3
Course Description	Introduction: 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. Probability Distributions: Discrete Probability Distributions, Continuous Probability Distributions. Fundamental Sampling Distributions: Sampling Distributions and Data Descriptions, Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of S^2, t-Distribution, F-Quantile and Probability Plots. Single Sample & One- and Two-Sample Estimation Problems: Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of P-Values for Decision Making in Testing Hypotheses (Single Sample & One- and Two-Sample Tests). Regression: 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.		
Text Book(s)	1. Dimitri P. Bertsekas, John Tsitsiklis, Introduction to probability, Athena Scientific, 2nd Edition, 2008, ISBN: 978-1886529236. 2. Jay L. Devore, Probability and Statistics for Engineering and the Sciences, Cengage Learning, 9th Edition, 2015, ISBN: 978-1305251809. 3. R.E. Walpole, R.H. Myers and S.L Myers, "Probability and Statistics for Engineers and Scientists", 9th Edition.		
Reference Material	1. MIT open courseware: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/		

Course Title	Pakistan Studies
Course Code	GE-166
Credit Hours	2 (2,0)
Category	General Education
Prerequisite	None
Co-Requisite	None
Follow-up	None
Course Description	Annexure A

Course Title	Fehm-e-Quran – I
Course Code	GE-167
Credit Hours	1 (0,1)
Category	General Education
Prerequisite	None
Co-Requisite	None
Follow-up	None
Course Description	Annexure B

Course Title	Fehm-e-Quran – II
Course Code	GE-168
Credit Hours	1 (0,1)
Category	General Education
Prerequisite	None
Co-Requisite	None
Follow-up	None
Course Description	Annexure B

Interdisciplinary / Allied Courses: Min 12 Credits

This section presents the mandatory Interdisciplinary and Allied Courses offered to students.

Course Title	Calculus and Analytical Geometry		
Course Code	AC-		
Credit Hours	3 (3,0)		
Category	Interdisciplinary / Allied Courses		
Prerequisite	None		
Co-Requisite	None		
Follow Up			
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Know the concepts and applications of calculus and analytical geometry.	C1 (Know)	1
	CLO2: Describe functions, limit, continuity chain rule and related techniques.	C2 (Describe)	1
	CLO3: Identify and solve problems related to differentiation and integration.	C3 (Apply)	1,3
	CLO4: Know the concepts analytical geometry.	C1 (Know)	1
Course Description	Motivation and applications of the course. Introduction to limits: Limits and Continuity, Techniques of finding limits, Indeterminate forms of limits, Introduction to functions: 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 Normal 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 R^3 , Equations for planes.		
Text Book(s)	1. Howard Anton, Irl C. Bivens and Stephen Davis, Calculus, 11 th Edition, Wiley, 2016, ISBN-10: 1119228581, ISBN-13: 978-1119228585.		
Reference Material	1. Thomas and Finney, Calculus and Analytic Geometry, 9 th Edition, ISBN-13: 978-0201531749, ISBN-10: 0201531747.		

Course Title	Linear Algebra		
Course Code	AC-		
Credit Hours	3 (3,0)		
Category	Interdisciplinary / Allied Courses		
Prerequisite	None		
Co-Requisite	None		
Follow-up	None		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	BT	PLO
	CLO1: Know the concept and application of leaner algebra.	C1 (Know)	1
	CLO2: Describe geometry of vector spaces and optimization.	C2 (Describe)	1
	CLO3: Perform matrix algebra.	C3 (Apply)	1,3
Course Description	Linear Equations in Linear Algebra: Systems of Linear Equations, Row Reduction and Echelon Forms, Vector Equations, The Matrix Equation $Ax = b$, Solution Sets of Linear Systems, Applications of Linear Systems, Linear Independence, Introduction to Linear Transformations, The Matrix of a Linear Transformation, Linear Models in Business, Science, and Engineering. Matrix Algebra: Matrix Operations, The Inverse of a Matrix, Characterizations of Invertible Matrices, Partitioned Matrices, Matrix Factorizations, Applications to Computer Graphics, Subspaces of R^n, Dimension and Rank. Determinants: Introduction to Determinants, Properties of Determinants, Cramer's Rule, Volume, and Linear Transformations. Vector Spaces: Vector Spaces and Subspaces, Null Spaces, Column Spaces, and Linear Transformations, Linearly Independent Sets; Bases, Coordinate Systems, The Dimension of a Vector Space, Rank, Change of Basis. Eigenvalues and Eigenvectors: Eigenvectors and Eigenvalues, The Characteristic Equation, Diagonalization, Eigenvectors and Linear Transformations, Complex Eigenvalues, Discrete Dynamical Systems. Orthogonality and Least Squares: Inner Product, Length, and Orthogonality, Orthogonal Sets, Orthogonal Projections, The Gram–Schmidt Process, Least-Squares Problems, Applications to Linear Models, Inner Product Spaces, Applications of Inner Product Spaces. Symmetric Matrices and Quadratic Forms: Diagonalization of Symmetric Matrices, Quadratic Forms, Constrained Optimization, The Singular Value Decomposition, Applications to Image Processing and Statistics. The Geometry of Vector Spaces: Affine Combinations, Affine Independence, Convex Combinations, Hyperplanes. Optimization: Matrix Games, Linear Programming—Geometric Method, Linear Programming—Simplex Method, Duality.		
Text Book(s)	- David C. Lay, Steven R. Lay, Judi J. McDonald, Linear Algebra and Its Applications, 5th Edition, Pearson, 2015, ISBN-13: 978-0321982384, ISBN-10: 032198238X. - Gilbert Strang, Introduction to Linear Algebra, 5th Edition, Wellesley-Cambridge Press, 2016, ISBN-13: 978-0980232776, ISBN-10: 0980232775. - Howard Anton, Elementary Linear Algebra, 11th Edition, Wiley, 2013, ISBN-13: 978-0470458211, ISBN-10: 0470458216.		
Reference Material	- Philip N. Klein, Coding the Matrix: Linear Algebra through Applications to Computer Science, 1st Edition, Newtonian Press, 2013, ISBN-13: 978-0615880990, ISBN-10: 0615880991. - David Hill, David Zitarelli, Linear Algebra Labs with MATLAB, 3rd Edition, Pearson, 2003, ISBN-13: 978-0131432741, ISBN-10: 0131432745.		