im | sciences

BACHELOR OF COMPUTER SCIENCE

PROGRAM OVERVIEW & CURRICULUM DETAILS

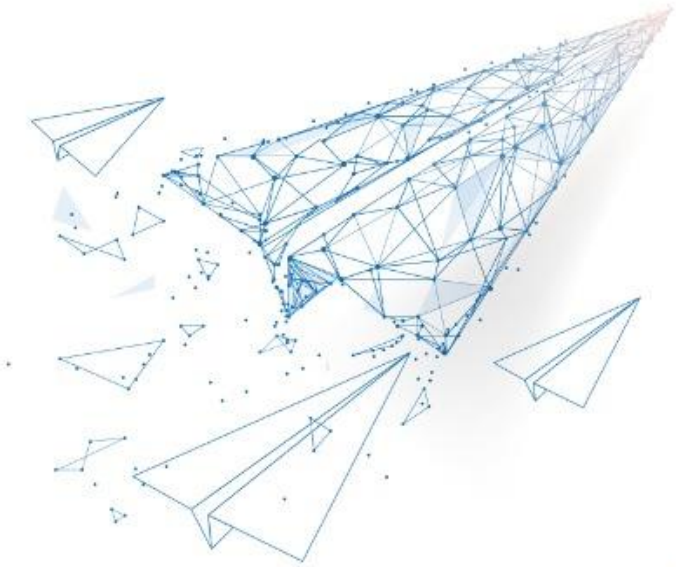


Table of Contents

1. Introduction

Computer Science is the study of the theoretical and practical aspects of computer technology and computer usage. Computer science major attracts students who are interested in both the mathematical theory and technical applications of computer science. While proficiency in computer programming is an essential skill, computer science students' deal with the fundamental nature of computation, what can and cannot be computed, and how efficiently a computation can be accomplished by a machine. They study how to build computer systems (computer architecture), how to make machines think (artificial intelligence), how to build computers that can see (computer vision), how to design software that works seamlessly from different locations (computer networks and distributed systems), how to model complex problems (modelling and simulation) and how to design more efficient computing algorithms (theory of computation).

The Center of Excellence in IT (CEIT) at IMSciences has highly research-active faculty, who encourage students to be involved in their applied/research work. Research opportunities focus both on fundamental research in theory of computation, computer networks, computer vision, data mining, software engineering, artificial intelligence, and robotics, as well as on projects of practical socio-economic significance, such as building systems to communicate relevant information to farmers, designing mapping solutions for rural areas and building record keeping solutions for the judicial system. A Computer Science (CS) degree is excellent preparation for the job market of the future and CS majors take up careers in every imaginable field. Our graduates have enjoyed excellent job placements, both within Pakistan and internationally. Many have chosen to make their own successful companies.

1.1 Program Structure

The Bachelor of Computer Science (BCS) combines a solid core of computer science courses as well as an understanding and appreciation for arts, humanities, and social sciences, and their importance in today's society. This is a 4-year degree program, which aims to enable students to apply scientific and engineering methodologies to the analysis of problems and the design, implementation, and evaluation of computer-based solutions. At the same instance, it attempts to utilize the breadth and depth of theoretical computer science and mathematics to emerging technologies and the ever-changing needs of industry as well as the cutting edge of computer science research.

1.2 Eligibility Criteria

- FA/F. Sc or Equivalent qualifications with at least second division, securing 50% marks in aggregate.
- The students who have not studied Mathematics at intermediate level must pass deficiency courses of Mathematics of 6 credit hours within one year of their regular studies.
- Qualifying the admission test and interview is compulsory. A candidate scoring less than 40% marks in the test and interview will stand disqualified for admission.
- Candidates who have secured at least 40% in the NTS-NAT are also eligible to apply.
- The merit of a candidate shall be measured by a 50 % weight age to the marks obtained in HSC or equivalent, 40 % to the marks obtained in the written test, and 10% to the marks obtained in the interview.
- A candidate shall be given a special credit of thirty marks for admission in each program mentioned above if he/she has studied Computer Science and/or statistics at intermediate level (for BS Data Science program only) at intermediate level or has done A level.
- The Hafiz Quran shall be given a special credit of 20 marks.
- The credit marks shall be added to the marks obtained at HSC or equivalent, subject to fulfilment of basic eligibility criteria of 50% marks.

1.3 Degree Requirements

For a BCS 4-year degree, a student is required to complete a minimum of 130-140 credit hours including a 6-credit hour research thesis/project. The normal duration for completion of BCS degree is 8 semesters over a period of 4 years. The maximum duration for obtaining BCS degree shall be 7 years.

1.4 Program Education Objectives (PEOs)

Following are the Program Education Objectives (PEOs) of BCS.

1. **(Fundamental Computing Knowledge)**: Graduates responsibly practicing in a variety of computer science and allied disciplines.
2. **(Basic Knowledge)**: Provide students with a combination of knowledge, hands-on experience, and application of theory to information issues.
3. **(Modern Tools usage)**: Graduates utilizing their skills and knowledge to solve complex computing problems in real world settings.
4. **(Continuous Improvement)**: Graduates demonstrating sustained learning and adapting to evolving fields through continued professional development and self-study.
5. **(Leadership/Teamwork):** A graduate who is effective in a leadership role of a team assigned to him/her or in an entrepreneurial environment.
6. **(Research Orientation):** Inculcate and enhance research abilities of students in respective fields.

1.5 Program Learning Outcomes (PLOs)[1] of BCS

1. Gain an understanding of the underpinning theories of fundamental principles and technologies within the area of computer science (**Academic education**).
2. 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 (**Knowledge for Solving Computing Problems**).
3. Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines (**Problem Analysis**).
4. 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 (**Design/ Development of Solutions**)
5. Create, select, adapt, and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations (**Modern Tool Usage**)
6. Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings (**Individual and Teamwork**)
7. Communicate effectively with the computing community and with society about complex computing activities by being able to comprehend and write effective reports,

[1] Adopted from Washington Accord

design documentation, make effective presentations, and give and understand clear instructions (**Communication**)

8. Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice (**Computing Professionalism and Society**)
9. Understand and commit to professional ethics, responsibilities, and norms of professional computing practice (**Ethics**)
10. Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional (**Life-long Learning**)

Upon completion of BCS degree, all the students should have attained the aforementioned ten PLOs/Student Outcomes (SOs).

1.6 Proposed Curriculum for BCS

Following are the proposed areas, which are required to cover to complete the degree. Covered areas consist of core courses (compulsory), foundation courses, general courses, and electives.

AREAS COVERED IN BCS

COMMON COURSES			
Course Group	**Min. No. of Credit Hours**	**Min. No. of Courses**	**Percentage**
General Education	19	7	14.2 %
Mathematics & Science Foundation	12	4	09.0 %
Computing Core	39	11	29.3 %
Institute Electives	12	4	09.0 %
Common Courses	**82**	**26**	**61.7 %**
DOMAIN COURSES			
Computer Science Core	24	7	18.0 %
Computer Science Electives	18	6	13.5 %
Computer Science Supporting	09	3	06.7 %
Domain Courses	**51**	**16**	**38.3 %**
TOTAL	**133**	**42**	**100%**

General Education Courses

Course Code	Course Title	Credit Hours	Contact Hours
CSC 301	Introduction to Information & Communication Technologies	3 (2-1)	2-3
ENG 301	English (General)	3 (3-0)	3-0
ENG 302	English (Functional)	3 (3-0)	3-0
ENG 401	English (Academic)	3 (3-0)	3-0
HSS 305	Fundamentals of Islamic Studies/ Ethics	2 (2-0)	2-0
HSS 301	Fundamentals of Pakistan Studies	2 (2-0)	2-0
CSC 595	Professional Practices	3 (3-0)	3-0
TOTAL		**19 (18-1)**	**18-3**

Mathematics and Science Foundation Courses

Course Code	Course Title	Credit Hours	Contact Hours
MTH 311	Calculus & Analytical Geometry	3 (3-0)	3-0
MTH 315	Linear Algebra	3 (3-0)	3-0
STA 415	Probability & Statistics	3 (3-0)	3-0
PHY 305	Applied Physics	3 (3-0)	3-0
TOTAL		**12 (12-0)**	**12-0**

Computing Core Courses

Course Code	Course Title	Credit Hours	Contact Hours
CSC 305	Programming Fundamentals	4 (3-1)	3-3
CSC 321	Discrete Structures	3 (3-0)	3-0
CSC 315	Object Oriented Programming	4 (3-1)	3-3
CSC 451	Database Systems	4 (3-1)	3-3
CSC 401	Data Structures & Algorithms	4 (3-1)	3-3
CSC 556	Information Security	3 (3-0)	3-0
CSC 575	Computer Networks	4 (3-1)	3-3
CSC 465	Operating Systems	4 (3-1)	3-3
SWE 401	Software Engineering	3 (3-0)	3-0
FYP 611	Final Year Project - I	3 (0-3)	0-9
FYP 612	Final Year Project - II	3 (0-3)	0-9
TOTAL		**39 (27-12)**	**27-36**

Institute Elective Courses

(Must be any FOUR courses or 12 credit hours, not limited to the areas listed below, Institutions may add/replace courses)

Course Code	Course Title	Credit Hours	Contact Hours
ACC 301	Fundamentals of Accounting	3 (3-0)	3-0
BUS 301	Introduction to Business	3 (3-0)	3-0
ENI 301	Entrepreneurship	3 (3-0)	3-0
FIN 301	Fundamentals of Business Finance	3 (3-0)	3-0
HRM 301	Fundamentals of Human Resource Management	3 (3-0)	3-0
HSS 311	Fundamentals of Sociology	3 (3-0)	3-0
HSS 415	Fundamentals of Psychology	3 (3-0)	3-0
HSS 505	Logic and Critical Thinking	3 (3-0)	3-0
MGT 301	Principles of Management	3 (3-0)	3-0
POL 301	Introduction to Political Science	3 (3-0)	3-0
POL 501	International Relations	3 (3-0)	3-0
LAN 512	Regional Language (Pashto)	3 (3-0)	3-0
LAN 513	Regional Language (Sindhi)	3 (3-0)	3-0
LAN 514	Regional Language (Punjabi)	3 (3-0)	3-0
LAN 521	Foreign Language (French)	3 (3-0)	3-0
LAN 522	Foreign Language (Chinese)	3 (3-0)	3-0
LAN 523	Foreign Language (German)	3 (3-0)	3-0
LAN 524	Foreign Language (Persian)	3 (3-0)	3-0
CSC 550	Computing and Society	3 (3-0)	3-0
TOTAL		**12 (12-0)**	**12-0**

Domain Courses for BCS
Computer Science CORE Courses

Course Code	Course Title	Credit Hours	Contact Hours
CSC 601	Artificial Intelligence	4 (3-1)	3-3
CSC 405	Digital Logic Design	4 (3-1)	3-3
CSC 531	Design and Analysis of Algorithms	3 (3-0)	3-0
CSC 411	Computer Organization & Assembly Language	4 (3-1)	3-3
SWE 539	Parallel and Distributed Computing	3 (3-0)	3-0
CSC 561	Compiler Construction	3 (3-0)	3-0
CSC 501	Theory of Automata	3 (3-0)	3-0
TOTAL		**24 (21-3)**	**21-9**

Computer Science SUPPORTING Courses
(Any Three from the list)

Coverage of relevant pre-requisite must be ensured while offering any of the following courses from this category

Course Code	Course Title	Credit Hours	Contact Hours
MTH 415	Multi-variate Calculus	3 (3-0)	3-0
CSC 510	Graph Theory	3 (3-0)	3-0
CSC 509	Theory of Programming languages	3 (3-0)	3-0
MTH 651	Numerical Computing	3 (3-0)	3-0
MTH 505	Differential Equations	3 (3-0)	3-0
TOTAL		**9 (9-0)**	**9-0**

Computer Science ELECTIVES Courses

(Must be any SIX courses or 18 credit hours, not limited to the areas listed below, Institute may add/replace courses)

Course Code	Course Title	Credit Hours	Contact Hours
CSC 611	Advance Database Systems	3 (3-0)	3-0
CSC 685	Machine Learning	3 (2-1)	2-3
CSC 605	Artificial Neural Networks	3 (2-1)	2-3
CSC 631	Cloud Computing	3 (3-0)	3-0
CSC 619	Text Mining	3 (3-0)	3-0
CSC 453	Fundamentals of Internet of Things (IoT)	3 (3-0)	3-0
CSC 461	Human Computer Interaction	3 (3-0)	3-0
CSC 505	Real-Time Systems	3 (3-0)	3-0
CSC 525	Computer Architecture	3 (3-0)	3-0
CSC 541	Systems and Network Administration	3 (2-1)	2-3
CSC 545	Computer Graphics	3 (2-1)	2-3
CSC 551	E-Commerce	3 (3-0)	3-0
CSC 565	Biometric Systems	3 (3-0)	3-0
CSC 571	Mobile Application Development	3 (3-0)	3-0
CSC 625	Natural Language Processing	3 (3-0)	3-0
CSC 641	Computer Vision	3 (2-1)	2-3
CSC 645	Wireless Networks	3 (3-0)	3-0
CSC 656	Multimedia Systems	3 (3-0)	3-0
CSC 675	Digital Image Processing	3 (2-1)	2-3
CSC 681	Fuzzy Logic	3 (3-0)	3-0
CSC 689	Expert Systems	3 (3-0)	3-0
SWE 411	Object Oriented Analysis and Design	3 (3-0)	3-0
SWE 515	Web Engineering	3 (2-1)	2-3
CSC 637	Selected Topics in IoT	3 (3-0)	3-0
TOTAL (Any SIX courses or 18 credit hours)		**18 (x-x)**	**x-x**

2. BCS – Semester-wise Breakdown

4-Year Program (8 Regular Semester of 18 weeks each) (133 Credit Hours)

Semester 1				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 301	Introduction to Information & Communication Technologies	3 (2-1)	2-3	
CSC 305	Programming Fundamentals	4 (3-1)	3-3	
ENG 301	English (General)	3 (3-0)	3-0	
HSS 301	Fundamental of Pakistan Studies	2 (2-0)	2-0	
MTH 311	Calculus and Analytical Geometry	3 (3-0)	3-0	
PHY 305	Applied Physics	3 (3-0)	3-0	
Total		**18(16-2)**	**16-6**	

Semester 2				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite(s)**
CSC 315	Object Oriented Programming	4 (3-1)	3-3	Programming Fundamentals
CSC 321	Discrete Structures	3 (3-0)	3-0	
ENG 302	English (Functional)	3 (3-0)	3-0	English (General)
HSS 305	Fundamentals of Islamic Studies	2 (2-0)	2-0	
MTH 315	Linear Algebra	3 (3-0)	3-0	
-	Institute Elective - I	3 (3-0)	3-0	
Total		**18(17-1)**	**17-3**	

Semester 3				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 401	Data Structures and Algorithms	4 (3-1)	3-3	Object Oriented Programming
CSC 405	Digital Logic Design	4 (3-1)	3-3	Applied Physics
ENG 401	English (Academic)	3 (3-0)	3-0	English (Functional)
-	Computer Science Supporting - I	3 (3-0)	3-0	
STA 415	Probability and Statistics	3 (3-0)	3-0	
Total		**17(15-2)**	**15-6**	

Semester 4				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 411	Computer Organization and Assembly Language	4 (3-1)	3-3	
CSC 451	Database Systems	4 (3-1)	3-3	Data Structures and Algo.
CSC 465	Operating Systems	4 (3-1)	3-3	Data Structures and Algo.
SWE 401	Software Engineering	3 (3-0)	3-0	
-	Institute Elective - II	3 (3-0)	3-0	
Total		**18(15-3)**	**15-9**	

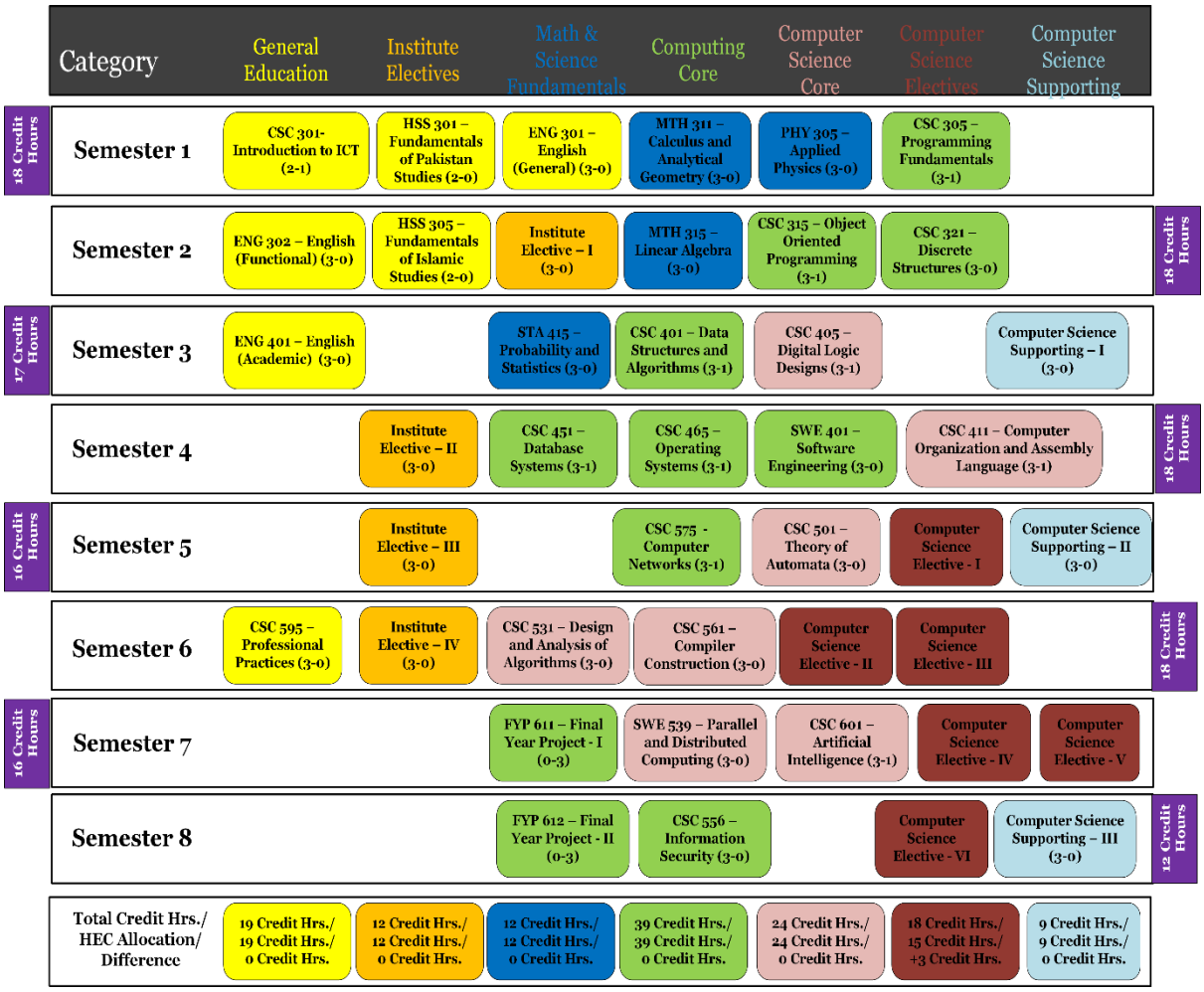
Semester 5				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 501	Theory of Automata	3 (3-0)	3-0	
CSC 575	Computer Networks	4 (3-1)	3-3	
-	Computer Science Supporting - II	3 (3-0)	3-0	
-	Institute Elective - III	3 (3-0)	3-0	
-	Computer Science Elective - I	3 (x-x)	x-x	
Total		**16(x-x)**	**x-x**	

Semester 6				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 531	Design and Analysis of Algorithms	3 (3-0)	3-0	Data Structures and Algo.
CSC 561	Compiler Construction	3 (3-0)	3-0	Theory of Automata
CSC 595	Professional Practices	3 (3-0)	3-0	
-	Computer Science Elective - II	3 (x-x)	x-x	
-	Institute Elective - IV	3 (3-0)	3-0	
-	Computer Science Elective - III	3 (3-0)	3-0	
Total		**18(x-x)**	**x-x**	

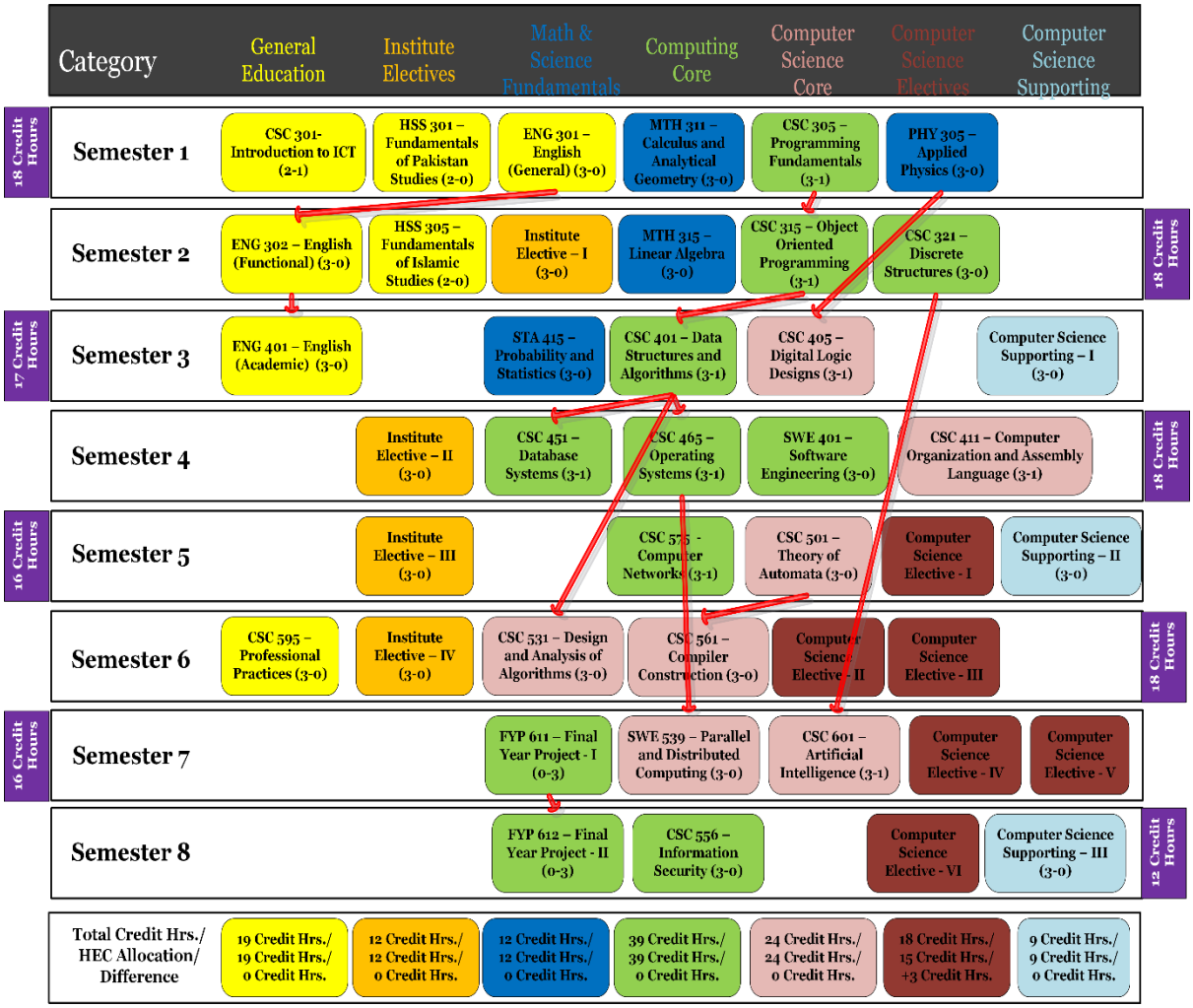
Semester 7				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 601	Artificial Intelligence	4 (3-1)	3-3	Discrete Structures
SWE 539	Parallel and Distributed Computing	3 (3-0)	3-0	Operating Systems
FYP 611	Final Year Project - I	3 (0-3)	0-9	
-	Computer Science Elective - IV	3 (x-x)	x-x	
-	Computer Science Elective - V	3 (x-x)	x-x	
Total		**16(x-x)**	**x-x**	

Semester 8				
Course Code	**Course Title**	**Credit Hours**	**Contact Hours**	**Pre-requisite**
CSC 556	Information Security	3 (3-0)	3-0	
FYP 612	Final Year Project - II	3 (0-3)	0-9	Final Year Project - I
-	Computer Science Supporting - III	3 (3-0)	3-0	
-	Computer Science Elective - VI	3 (x-x)	x-x	
Total		**12(x-x)**	**x-x**	

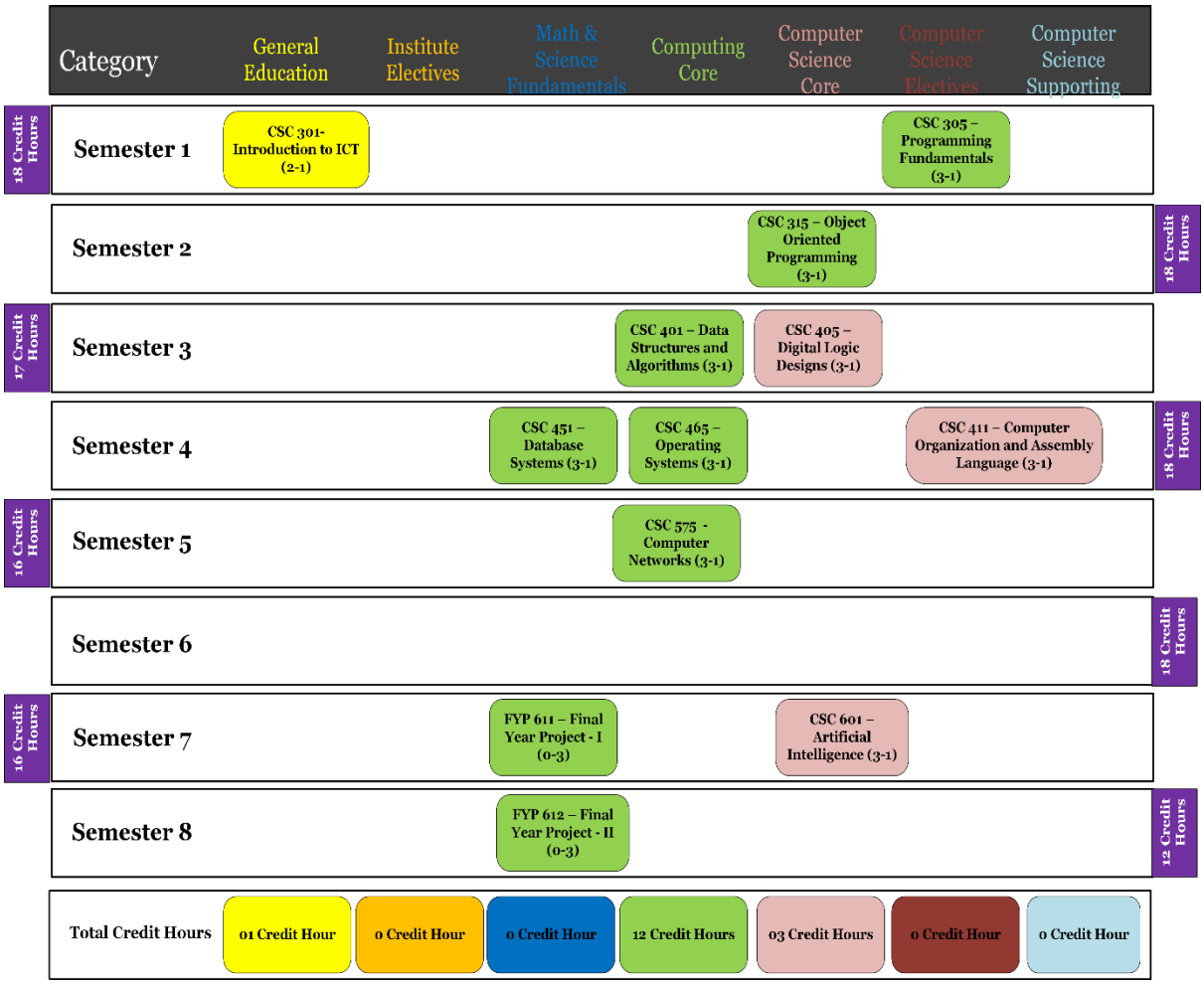
2.1 Detail overview of courses in BCS



2.2 Dependency Graph for Courses of BCS



2.3 Distribution of Labs for Courses of BCS



3. Course Outlines

Following are the course outlines of all the courses in BCS.

Introduction to Information & Communication Technology			
Credit Hours	3 (2-1)	**Prerequisites**	None

Course Introduction:

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

Course Objectives

Upon successful completion of a major in Introduction to ICT, students will be able to;

- Demonstrate proficiency in problem-solving techniques using the computer.
- Identify and describe major hardware components, basics of storage devices, number systems, machine cycle, microcomputer processor and use communications and networking terminology further include Internet operations and its uses.
- Students will be able to develop understanding of Computer programming is by its nature inherently mathematical.
- Learning programming language is challenging and difficult and hard work for most students but upon completing this course the students should be able to understand the basic concepts related to programming.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand basics of computing technology	C1	Knowledge
2. Do number systems conversions and arithmetic	C2	Understanding
3. Have knowledge of types of software	C2	Understanding
4. Have knowledge of computing related technologies	C3	Apply

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

Course Content:

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

Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Charles S. Parker, Understanding Computers: Today and Tomorrow, Course Technology, 25 Thomson Place, Boston, Massachusetts 02210, USA 16th Edition 2. Livesley, Robert Kenneth. An introduction to automatic digital computers. Cambridge University Press, 2017. 3. Zawacki-Richter, Olaf, and Colin Latchem. "Exploring four decades of research in Computers & Education." Computers & Education 122 (2018): 136-152.

English General			
Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

General English and Intensive English courses are designed to help students make rapid progress in English and focus on the four key language skills – reading, writing, listening, and speaking – with lots of additional work on vocabulary, grammar, and pronunciation.

Course Objectives:

- To evaluate information and its sources critically.
- To incorporate selected information into one's knowledge base.
- To use information effectively to accomplish a specific purpose

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Enrich the thought and culture and provides us with the most important international vehicle of expression.	C1	Remember
2. Enhance English language skills of the students and develop their critical thinking.	C3	Apply
3. Demonstrate ability to think critically	C3	Apply

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

Course Content:

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

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Final Exam

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000 2. Comprehensive objective general English: For all competitive exams (with practice sets): For Bank (PO/MT/Clerk), RBI, LIC, SSC (CGL, MTS, LDC), UPSC, IES, SCRA, RRB, NDA, cds, Armed Forces, MBA, Nift Jimper, Hotel Management, MCA, Clat, CTET, B. Ed. & other examinations (2016). New Delhi: Source Books a unit of Viva Books Private Limited. 3. Rivers, W. M. and M.S Temperley. 1978. A Practical Guide to the Teaching of English as a Second or Foreign Language. Oxford University Press, Oxford, UK. 4. Smalley, R. L., M. K Ruetten and D. Kozyrev. 2001. Refining Composition Skills. 4th Ed. Heinle & Heinle Inc., Boston, MA, USA. 5. Vawdrey C. 1993. Practical Business English. 2nd Ed. Richard d Irwin Publishing, New York City, NY, USA.

English Functional

Credit Hours	3 (3-0)	**Prerequisites**	English General

Course Introduction:

Functional English is usage of the English language required to perform a specific function. This is typically taught as a foundation subject when a good command of English is required for academic study and career progression.

Course Objectives:

- Strengthen the language skills in order for the students to use language effectively as a tool to succeed in academic activities which they will be carrying out as part of their academic activities.
- Enhance the development of all the four language skills but explicitly focuses on listening, reading, and writing; and the efforts made in these areas are perceived to implicitly target proficiency and accuracy in the target language, English. The language skills are coincided with study skills which are directly required by students as basic skills to pursue other subjects more meaningfully

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Deliver effective presentations and participate actively in group discussions	C3	Apply
2. Complete Academic Writing tasks using writing process and strategies according to genres	C5	Evaluate
3. Use Language Skills and Strategies in different situations, for a variety of functions	C5	Evaluate

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

Course Content:

Basics of Grammar, Parts of speech and use of articles, Sentence structure, Active and passive voice, Practice in unified sentence, Analysis of phrase, Clause and sentence structure, Transitive and intransitive verbs, Punctuation and spelling, Comprehension, Answers to questions on a given text, General topics and everyday conversation, Translation skills (Urdu to English), Paragraph writing, Presentation skills, Extensive reading is required for vocabulary building

Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Writing. Intermediate by Marie-Christine Boutin, Suzanne Brinand and Francoise Grellet. Oxford Supplementary Skills. Fourth Impression 1993. ISBN 0 19 435405 7 Pages 20-27 and 35-41. 2. Reading. Upper Intermediate. Brain Tomlinson and Rod Ellis. Oxford Supplementary Skills. Third Impression 1992. ISBN 0 19453402 2. 3. Practical English Grammar by A. J. Thomson and A. V. Martinet. Exercises 1. Third edition. Oxford University Press. 1997. ISBN 0194313492 4. Practical English Grammar by A. J. Thomson and A. V. Martinet. Exercises 2. Third edition. Oxford University Press. 1997. ISBN 0194313506

English Academic

Credit Hours	3 (3-0)	**Prerequisites**	English Functional

Course Introduction:

English for Academic Purposes (EAP), commonly known as Academic English, entails training students, usually in a higher education setting, to use language appropriate for study. It is one of the most common forms of English for Specific Purposes (ESP).

Course Objectives:

The primary objectives for this course are to:

- Interact with academic content: reading, writing, listening, and speaking.
- Demonstrate ability to think critically.
- Utilize information and digital literacy skills

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Interact with academic content: reading, writing, listening, and speaking.	C1	Knowledge
2. Demonstrate ability to think critically.	C3	Apply
3. Utilize information and digital literacy skills.	C3	Apply

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

Course Content:

Principles of writing good English; understanding the composition process: writing clearly; words: sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation; Process of writing; observing, audience collecting: composing, drafting and revising: persuasive writing: reading skills: listening skills and comprehension: skills for taking notes in class: skills for exams; Business communications; planning messages: writing concise but with impact: Letter formats; mechanics of business: letter writing: letters: memo and applications; summaries: proposals: writing resumes: styles and formats: oral communications: verbal and

non-verbal communication: conducting meetings; small group communication: taking minutes: Presentation skills; presentation strategies: defining the objective: scope and audience of the presentation: material gathering material organization strategies: time management; opening and concluding: use of audio-visual aids: delivery and presentation.
Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748 2. Smalley, R. L., M. K Ruetten and D. Kozyrev. 2001. Refining Composition Skills. 4th Ed. Heinle & Heinle Inc., Boston, MA, USA. 3. Schriver, K. A. 1997. Dynamics in Document Design. 3rd Ed. Wiley Inc. New York City, NY, USA. 4. Henri, E. B., C. J. Jacobs, K. G. Langendoen and D. Grune. 2012. Modern Compiler Design. 2nd Ed, John Wiley & Sons. New York City, NY, USA. 5. Masami, I. 2004. Algebraic Theory of Automata and Languages. World Scientific, River Edge, NJ, USA.

Fundamentals of Islamic Studies

Credit Hours	2 (2-0)	**Prerequisites**	None

Course Introduction:

An Introduction to the academic understanding of Islam. Topics may include Faith, rituals, law (Shari'ah), jurisprudence (Fiqh), theology (Kalam), and stories from the Islamic heritage. Non-Western multicultural course.

Course Objectives:

This course will:

- Enable the learners to develop knowledge and interest towards Shariah, Quran, and Hadith.
- Assist the learners in character building and to develop Islamic approach & thinking amongst the students.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain the basic concepts of Shariah, Quran, and Hadith.	C2	Outlook towards profession, ethics, and society
2. Demonstrate the Islamic approach and thinking through their positive and religious character	C3	Reflection and critical thinking skills

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

Course Content:

Basic Concepts of Quran, History of Quran, Uloom-ul –Quran, Verses of Surah Al-Baqra Related to Faith(Verse No-284-286), Verses of Surah Al-Hujrat Related to Adab Al-Nabi(Verse No-1-18), Verses of Surah Al-Mumanoon Related to Characteristics of faithful (Verse No-1-11), Verses of Surah al-Furqan Related to Social Ethics (Verse No.63-77), Basic Concepts of Hadith, History of Hadith, Kinds of Hadith, Uloom –ul-Hadith, Sunnah & Hadith, Legal Position of Sunnah, Basic Concepts of Islamic Law & Jurisprudence, History & Importance of Islamic Law & Jurisprudence, Sources of Islamic Law & Jurisprudence, Nature of Differences in Islamic Law, Basic Concepts of Islamic Culture & Civilization, Historical Development of Islamic Culture & Civilization, Characteristics of Islamic Culture & Civilization, Islamic Culture & Civilization and Contemporary Issues

Teaching Methodology:

Lecturing, Written Assignments, Final Exam

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI
2. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services
3. Waliullah M., 1982. Muslim Jurisprudence and the Quranic Law of Crimes. 2nd Ed. Islamic Book Service, Karachi, Pakistan

Fundamentals of Pakistan Studies

Credit Hours	2 (2-0)	**Prerequisites**	None

Course Introduction:

Pakistan Studies is the integrated, coordinated, and systematic area of study that draws upon various social science disciplines such as history, geography, anthropology, economics, political science, and sociology in relation to Pakistan. It is one of the compulsory courses at the secondary school and higher secondary school levels of education. The social science departments of many universities offer it as a degree course, but there are also university departments dedicated to the education and research in Pakistan Studies.

Course Objectives:

The course aims to:

- Familiarize the students to their past and present, focusing on the history and ideology of Pakistan, its contemporary issues and foreign policy.
- Inculcate in students the sense of belonging to Pakistan in order to make them useful members of the society who can benefit the country by expanding developments in different fields.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Demonstrate the basic knowledge of the historical and ideological perspectives of Pakistan, its current challenges, and its relationship with the neighboring countries.	C2	Individual and Teamwork
2. Identify the role of different systems, treaties and conventions established to cater human rights at national and international level.	C4	Life-long Learning

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

Course Content:

Ideological rationale with special reference to Sir Syed Ahmed Khan, Allama Muhammad Iqbal and Quaid-i-Azam Muhammad Ali Jinnah., Factors leading to Muslim separatism, People and Land, Indus Civilization, Muslim advent, Location and Geo-Physical features, Economic institutions and issues, Society and social structure, Ethnicity, Foreign policy of Pakistan and challenges, Futuristic outlook of Pakistan, Political and constitutional phases: (1947-58, 1958-71, 1971-77, 1977-88, 1988-99, 1999 onward).

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Final Exam

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. The making of Pakistan, Aziz. 1976
2. A Short History of Pakistan, I. H. Qureshi, ed., Karachi, 1988
3. Mehmood, S. 1994. Pakistan Political Roots and Development. 2nd Ed. Five Star Publishing, Lahore, Pakistan.
4. S.M. Burke and L. Ziring. 1993. Pakistan's Foreign Policy: An Historical Analysis. 2nd Ed. Oxford University Press, Oxford, U.K.

Professional Practices

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

Professional Practice is a term used to describe activities, which will help you apply your knowledge to your industry, job role or workplace.

Course Objectives:

The primary objectives are:

1. Introduce the basic concepts and importance of ethics that can be mapped in the professional lives.
2. Highlight the Impact of social media and social implications of computing and networked communication regarding ethics and morality
3. The making and implementation of framework for ethical decision making
4. An understanding of professional ethical theories and code of ethics (IEEE/ACM)
5. Demonstrate the concepts of intellectual property and privacy, their rights, laws, and their types
6. Highlight the concepts of anonymity, security policies, computer crimes, social engineering, and to provide the guidelines for a sustainable practitioner.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Know the scope of computing field after graduating in it and what are the common things in every organization.	C1	Knowledge
2. Distinguish between various fields of computing.	C2	Problem Solving
3. Describe the core of any profession.	C3	Understanding
4. Write and analyze software contracts as an employer or to an employer.	C3	Analysis
5. Know the business and professional environment of software house.	A2	Ethics

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

Course Content:

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and Accounting, Anatomy of a Software House, Computer Contracts, 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, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Final Exam

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Habash, R. (2019) Professional practice in engineering and Computing: Preparing for future careers. Boca Raton: CRC Press.
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition (January 3, 2009). ISBN10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet (3rd Edition) by Sara Baase, Prentice Hall; 3rd Edition (2008). ISBN-10: 0136008488

Calculus & Analytical Geometry

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

This freshmen level course has been designed to introduce the ideas and concepts of Calculus that would serve as a foundation for subsequent computer engineering courses. The primary objective is to endow the knowledge of basic concepts of calculus and geometry. Purpose of this course is to build the student's knowledge of differential/integral calculus of multi-variable functions based on their experience of differential/integral calculus and analytic geometry of functions of one independent variable, at the Intermediate level.

Course Objectives:

- The primary objective is to endow the knowledge of basic concepts of calculus and geometry.
- Purpose of this course is to build the student's knowledge of differential/integral calculus of multivariable functions based on their experience of differential/integral calculus and analytic geometry of functions of one independent variable, at the Intermediate level.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Comprehend key concepts of single variable calculus, differential calculus, integral, multivariate calculus, and analytical geometry.	C2	Understanding
2. Apply the fundamentals of functions, limits and continuity, derivative, integration, Partial differentiation to engineering problems.	C3	Knowledge
3. Solve problems of analytical geometry using rectangular co-ordinates systems in 3 dimensions.	C3	Problem Solving

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

Course Content:

Limits and Continuity; Introduction to functions, Introduction to limits, Techniques of funding limits, Indeterminate forms of limits, Continuous and discontinuous functions and their applications, Differential calculus; Concept and idea of differentiation, Geometrical and Physical meaning of derivatives, Rules of differentiation, Techniques of differentiation, Rates of change, Tangents and 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 R3, Equations for planes

Teaching Methodology:

Lecturing, Written Assignments
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Calculus and Analytic Geometry by Kenneth W. Thomas. 2. Calculus by Stewart, James. 3. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole.

Linear Algebra

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

This Course covers matrix theory and linear algebra emphasizing topics useful in other disciplines is a requirement for mathematics, and it's highly recommended for engineering majors. Topics include systems of linear equations and their solutions, matrices and matrix algebra, inverse matrices; determinants; real n-dimensional vector spaces, abstract vector spaces and their axioms, linear transformation; dot/ cross products, Subspaces, linear independence, bases for vector spaces, dimension, matrix rank, eigenvectors, eigenvalues, and matrix diagonalization. Some applications of linear algebra will be discussed, such as Kirchhoff's laws.

Course Objectives:

- The main objective of this course is to help students learn in rigorous manner, the tools and methods essential for studying the solution spaces of problems in mathematics, engineering, the natural sciences, and social sciences and develop mathematical skills needed to apply these to the problems arising within their field of study; and to various real-world problems.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Apply the basic operation of matrix algebra.	C3	Application
2. Demonstrate the concepts of two and three-dimensional geometry.	C3	Understanding
3. Discuss the area, volumes of bounded regions by using multiple integrals.	C3	Knowledge

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

Course Content:

System of Linear Equations and Matrices, Introduction to system of linear equations, Matrix form of system of Linear Equations, Gaussian Elimination method, Gauss-Jorden Method, Consistent and inconsistent systems, Homogeneous system of equations, Vector Equations, Introduction to vector in plane, Vector form of straight line, Linear Combinations, Geometrical interpretation of solution of Homogeneous and Non-homogeneous equations, Applications of Linear Systems, Traffic Flow Problem, Electric circuit Problem, Economic Model, Linear transformations, Introduction to linear transformations, Matrix transformations, Domain and range of linear transformations, Geometric interpretation of linear transformations, Matrix of linear transformations, Inverse of a matrix, Definition of inverse of a matrix, Algorithm to find the inverse of matrices, LU factorization, Introduction to determinants, Geometric meaning of determinants, Properties of determinants, Crammer Rule, Cofactor method for finding the inverse of a matrix, Definition of vector spaces,

Subspaces, Spanning set, Null Spaces and column spaces of linear transformation, Linearly Independent sets and basis, Bases for Null space and Kernel space, Dimension of a vector space, Introduction to Eigen value and Eigen vectors, Computing the Eigen values, Properties of Eigen values, Diagonalization, Applications of Eigen values.
Teaching Methodology:
Lecturing, Written Assignments
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Elementary Linear Algebra by Howard Anton 2. Linear Algebra and its Applications by Gibert Strang

Probability & Statistics			
Credit Hours	3 (3-0)	**Prerequisites**	None
Course Introduction:			
This course introduces probability and statistics with applications. Topics include basic probability models; combinatory; random variables; discrete and continuous probability distributions; statistical estimation and testing; confidence intervals; and an introduction to linear regression.			
Course Objectives:			
• The successful completion should develop understanding of the systems which involve uncertainty. • Further, it should lay down the analyzing and evaluating techniques for these systems.			
Course Learning Outcomes (CLOs):			
At the end of the course the students will be able to:		**Domain**	**BT Level***
On completion of this course, the student will be able to: 1. Explain the basic concept of Statistics and Probability and their need in engineering/Science. 2. Analyze random variables, probability distributions and sampling distributions. 3. Apply different probability and statistics techniques in engineering problems		C2 C4 C3	Explanation Analyze Apply
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain			
Course Content:			
Introduction to Statistics and Data Analysis, Statistical Inference, Samples, Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule. Random Variables and Probability Distributions. Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem.			

Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of S2, t-Distribution, 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), 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.
Teaching Methodology:
Lecturing, Written Assignments, Presentation, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam
Reference Materials:
1. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116 2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, Duxbury Press; 3rd Edition (February 3, 2006), ISBN-10:0495107573 3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259

Applied Physics

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

The course covers topics in Physics that are directly related to Mechanical Engineering like Mechanics, Electromagnetic waves, Alternating current circuits and solid-state physics.

Course Objectives:

- Understanding of the fundamental concepts/laws in physics by explaining and discussing the physics as well as their relevance to everyday events and circumstances in a broad interdisciplinary context.
- Demonstrate teamwork skills/ ability to collaborate by working in groups on a laboratory experiment
- Reveal critical thinking/ analytical reasoning ability by setting up mathematical descriptions of physical systems and to calculate measurable quantities that provide an understanding of the physical environment in terms of the concepts listed in the course content.
- Ability to apply knowledge/skills to real world settings

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Define how to calculate and measure Voltage, Current and Resistance, connectivity etc. using digital mustimeter and express knowledge of handling Power Trainer, Function Generator and Oscilloscope	P1	Knowledge
2. Use the knowledge acquired in lab and course to construct and investigate basic electronic circuit like dc power supply to harvest knowledge of all its intermediate stages	C6	Understanding

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

Course Content:

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, Toroid's, 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.

Teaching Methodology:

Lecturing, Written Assignments, Project, Experiments, Report Writing

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Report Writing, Experiments, Final Exam

Reference Materials:

1. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker
2. Ntalianis, K. et al. (2019) Applied Physics, System Science and Computers III: Proceedings of the 3rd international conference on applied physics, system science and computers (APSAC2018), September 26-28, 2018, Dubrovnik, Croatia. Cham: Springer.
3. Narciso Garcia, Arthur Damask, Steven Schwarz., "Physics for Computer Science Students", Springer Verlag, 1998

Programming Fundamentals

Credit Hours	4 (3-1)	**Prerequisites**	None

Course Introduction:

Programming is an increasingly important skill, whether you aspire to a career in software development, or in other fields. This course is the first in the specialization Introduction to Programming in C, but its lessons extend to any language you might want to learn. This is because programming is fundamentally about figuring out how to solve a class of problems and writing the algorithm, a clear set of steps to solve any problem in its class. This course will introduce you to a powerful problem-solving process—the Seven Steps—which you can use to solve any programming problem. In this course, you will learn how to develop an algorithm, then progress to reading code and understanding how programming concepts relate to algorithms.

Course Objectives:

The objective of course is to;

- Introduce a disciplined approach to Problem solving methods and algorithm development.
- Teach the syntax and vocabulary of a modern programming language like C++. The significant philosophies and logical programming, including models for I/O, processing, and all related terminology will be taught. Simple programs will be constructed, using a number of different logical, calculation and algorithm.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand basic problem-solving steps and logic constructs	C2	Understanding
2. Apply basic programing concepts	C3	Apply
3. Design and implement algorithms to solve real world problems.	C6	Create

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

Course Content:

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

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Final Exam

Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam
Reference Materials:
1. Starting out with Python, 6th Edition, Tony Gaddis. 2. Starting out with Programming Logic & Degins, 6th Edition, Tony Gaddis, 3. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie 4. Object Oriented Programming in C++ by Robert Lafore 5. Introduction to Computation and Programming Using Python: With Application to Understanding Data, Latest Edition by Guttag, John

Discrete Structures

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

Discrete Structures is the study of objects that have discrete as opposed to continuous values including the foundations of logic, algorithms and their complexity, mathematical reasoning, relations, graphs, trees and combinatorics.

Course Objectives:

By the end of the course the students will be able to:

1. To design hardware circuits by using gates.
2. To convert expressional statement into mathematical models.
3. To apply the knowledge and skills obtained to investigate and solve a variety of discrete mathematical problems.
4. To produce convincing argument, conceive and/or analyze basic mathematical proofs and discriminate between valid and unreliable arguments.
5. To make effective use of appropriate technology using graphs, trees, and relations in computer science problems (Data Base, Artificial intelligence, Game Theory, Algorithm Analysis)

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs, and Trees etc.	C2	Understanding
2. Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.	C3	Apply
3. Apply discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.	C3	Apply
4. Differentiate various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular.	C4	Analyze

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

Course Content:

Mathematical reasoning, propositional and predicate logic, rules of inference, proof by induction, proof by contraposition, proof by contradiction, proof by implication, set theory, relations, equivalence relations and

partitions, partial orderings, recurrence relations, functions, mappings, function composition, inverse functions, recursive functions, Number Theory, sequences, series, counting, inclusion and exclusion principle, pigeonhole principle, permutations and combinations, elements of graph theory, planar graphs, graph coloring, Euler graph, Hamiltonian path, rooted trees, traversals.
Teaching Methodology:
Lectures, Written Assignments, Practical labs, Semester Project, Presentations
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam
Reference Materials:
1. Discrete Mathematics and Its Applications, 8th edition by Kenneth H. Rosen 2. Discrete Mathematics with Applications, 5th Edition by Susanna S. Epp 3. Discrete Mathematics, global edition by Richard Johnson Baugh 4. Discrete Mathematical Structures, 7th edition by Kolman, Busby & Ross 5. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi 6. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred Grassman

Object Oriented Programming			
Credit Hours	4 (3-1)	**Prerequisites**	Programming Fundamentals
Course Introduction:			
This course introduces advanced programming skills and focuses on the core concepts of object-oriented programming and design using a high-level language, either Python or Java. Object-oriented programming represents the integration of software components into a large-scale software architecture. Software development in this way represents the next logical step after learning coding fundamentals, allowing for the creation of sprawling programs. The course focuses on the understanding and practical mastery of object-oriented concepts such as classes, objects, data abstraction, methods, method overloading, inheritance, and polymorphism. Practical applications in the domain of data science and as seen in stacks, queues, lists, and trees will be examined.			
Course Objectives:			
At the end of the course, the students will be able to: • Explain the steps in creating an executable program for a computer, including the intermediate representations and their purpose. • Apply good programming style and understand the impact of style on developing and maintaining java programs. • Explain the benefits of object-oriented design and understand when it is an appropriate methodology to use for java programming. • Design object-oriented solutions for small systems involving multiple objects. • Implement solutions in Java and exception handling techniques. • Working with methods overloading, passing arguments to objects, returning objects and constructors. • Explain the relevance of ethics in the context of Software Engineering.			
Course Learning Outcomes (CLOs):			

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand principles of object-oriented paradigm.	C2	Understanding
2. Identify the objects & their relationships to build object-oriented solution	C3	Apply
3. Model a solution for a given problem using object-oriented principles	C3	Apply
4. Examine an object-oriented solution.	C4	Analyze

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

Course Content:

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

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis
2. C++ How to Program, 10th Edition, Deitel & Deitel.
3. Object Oriented Programming in C++, 5th Edition by Robert Lafore
4. Java: How to Program, 10th Edition by Paul Deitel
5. Beginning Java 2, 8th Edition by Ivor Horton
6. An Introduction to Object Oriented Programming with Java, 7th Edition by C. Thomas Wu

Database Systems

Credit Hours	4 (3-1)	**Prerequisites**	Data Structure and algorithms

Course Introduction:

A study of database models including the hierarchical, network, relational and object-oriented models, and the examination of such practical issues as database design, setup, and manipulation. Other selected topics include data integrity, data security, backup and recovery procedures, database administration, etc. Several programming projects are assigned involving the use of a database management system.

Course Objectives:

The main objective of this course is to provide students with the background to design, implement, and use database management systems. After the completion of this course students will be able to:

- Model and design Database
- Write Structured Queries and optimize them
- Implement Constraints and Triggers
- Use and develop semi structured databases

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain fundamental database concepts.	C2	Understanding
2. Design conceptual, logical, and physical database schemas using different data models.	C5	Evaluate
3. Identify functional dependencies and resolve database anomalies by normalizing database tables.	C2	Understanding
4. Use Structured Query Language (SQL) for database definition and manipulation in any DBMS	C4	Analyze

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

Course Content:

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

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Database Systems: A Practical Approach to Design, Implementation, and Management, 7th Edition by Thomas Connolly and Carolyn Begg
2. Database Systems: The Complete Book, 5th Edition by Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom

3. Database System Concepts, 7th Edition by Avi Silberschatz, Henry F. Korth and S. Sudarshan.

Database Management Systems, 7th Edition by Raghu Ramakrishnan, Johannes Gehrke

Data Structures and Algorithms

Credit Hours	4 (3-1)	**Prerequisites**	Object Oriented programing

Course Introduction:

An overview of data structure concepts, arrays, stack, queues, trees, and graphs. Discussion of various implementations of these data objects, programming styles, and run-time representations. Course also examines algorithms for sorting, searching and some graph algorithms. Algorithm analysis and efficient code design is discussed.

Course Objectives:

At the end of the course, the students will be able to:

- Introduce the concept of data structures and algorithms
- Understand and use various efficient storage mechanisms of data for an easy access in a program development.
- Design and implement various basic and advanced data structures.
- Understand and use Searching and Sorting techniques.
- Develop applications using efficient data structures like Stacks, Queues, Lists, Graphs and Trees.
- Demonstrate the concept of protection and management of data.
- Improve the logical ability by writing algorithms and systematic approach in solving problems with the help of a suitable data structure.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Implement various data structures and their algorithms and apply them in implementing simple applications.	C2,3	Understanding, Apply
2. Analyze simple algorithms and determine their complexities.	C4,5	Analyze, Evaluate
3. Apply the knowledge of data structures to other application domains.	C3	Apply
4. Design new data structures and algorithms to solve problems.	C6	Create

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

Course Content:

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

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam
Reference Materials:
1. Data Structures and Algorithms in C++ by Adam Drozdek 2. Data Structures and Algorithm Analysis in Java by Mark A. Weiss 3. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry 4. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss 5. Java Software Structures: Designing and Using Data Structures by John Lewis and Joseph Chase

Information security			
Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

The subject of computer networking is enormously complex, involving many concepts, protocols, and technologies. To cope with the scope and complexity these protocols and technologies are woven together in an intricate manner in what is called the layered protocol stack (or suite). The layered organization allows breaking down complex functions required for computers networking into manageable tasks. This course is an introduction to computer networking using a top-down approach—that is, by beginning at the highest layer of the protocol stack (application layer) and proceeding down through different layers towards the lowest one (the physical layer). The course places emphasis on the application layer (a "high growth area" in networking). The course uses the Internet's architecture and protocols as the primary vehicle for studying fundamental computer networking concepts. More than often, the course will also include concepts and protocols from other network architectures. But the main focus is on the Internet, a fact reflected in organizing the course around the Internet's five-layer architecture.

Course Objectives:

By the end of this course the students will be able to:

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
- Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain key concepts of information security such as design principles, cryptography, risk management, and ethics	C2	Explain
2. Discuss legal, ethical, and professional issues in information security	A2	Discuss
3. Apply various security and risk management tools for achieving information security and privacy	C3	Apply
4. Identify appropriate techniques to tackle and solve problems in the discipline of information security	C4	Identify

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain
Course Content:
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.
Teaching Methodology:
Lectures, Written Assignments, Semester Project, Presentations
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Whitman, M.E. and Mattord, H.J. (2022) Principles of Information Security. Boston, MA: Cengage. 2. Computer Security: Principles and Practice, 4th edition by William Stallings 3. Principles of Information Security, 8th edition by M. Whitman and H. Mattord 4. Computer Security, 3rd edition by Dieter Gollmann 5. Computer Security Fundamentals, 4th edition by William Easttom 6. Official (ISC)2 Guide to the CISSP CBK, 5th edition

Computer Networks

Credit Hours	4 (3-1)	**Prerequisites**	None

Course Introduction:

This course is to provide students with an overview of the concepts and fundamentals of data communication and computer networks.

Course Objectives:

By the end of the course, the students will be to:

1) Understand the TCP/IP protocol suite and the working of the Internet.
2) Form an understanding of the principles upon which the global Internet was designed.
3) Understand basic terminology so that students can understand networking research papers.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe the key terminologies and technologies of computer networks	C2	Describe
2. Explain the services and functions provided by each layer in the Internet protocol stack.	C2	Explain
3. Identify various internetworking devices and protocols and their functions in a networking	C1	Identify
4. Analyze working and performance of key technologies, algorithms, and protocols	C4	Analyze
5. Build Computer Network on various Topologies	P3	Build

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

Course Content:

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

Teaching Methodology:

Lecturing, Written Assignments, Project, lab tasks

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Computer Networking: A Top-Down Approach Featuring the Internet, 8th edition by James F. Kurose and Keith W. Ross
2. Computer Networks, 5th Edition by Andrew S. Tanenbaum
3. Data and Computer Communications, 11th Edition by William Stallings
4. Data Communication and Computer Networks, 7th Edition by Behrouz A. Forouzan

Operating Systems

Credit Hours	4 (3-1)	**Prerequisites**	Programming Fundamentals, Data Structure and Algorithms

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 Objectives:

Make the students be able to:

1. Build an understanding about the fundamental concepts of operating systems.
2. Know about the structure of an operating system, its components, design strategies, algorithms and schemes used to design and implement different components of an operating system
3. Familiarize with the basic taxonomy and terminology of operating systems.
4. Study any advance courses that involve operating system concepts.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems	C2	Understanding
2. Identify the core functions of operating systems and how they are architected to support these functions,	C1	Identify
3. Analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions	C5	Evaluate
4. Demonstrate the knowledge in applying system software and tools available in modern operating systems.	C3	Apply

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

Course Content:

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

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Operating Systems Concepts, 10th edition by Abraham Silberschatz
2. Modern Operating Systems, 5th edition by Andrew S. Tanenbaum
3. Operating Systems, Internals and Design Principles, 9th edition by William Stallings

Software Engineering

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

This course introduces students to the different software development lifecycle (SDLC) phases used in developing, delivering, and maintaining software products. Students will also acquire basic software development skills and understand common terminology used in the software engineering profession.
Students will also learn and practice using traditional coding standards/guidelines. Python software development libraries and debugging tools will be explored and used in projects to familiarize students with basic tasks involved in modifying, building, and testing software. The course will also lay the foundation for achieving academic and career success in Software Engineering.

Course Objectives:

During this course, students will be able to:

- List and describe the fundamental phases of the Software Development Lifecycle (SDLC)
- Define and describe fundamental software engineering terminology and coding practices
- Explore/explain relationships between software engineering and other engineering disciplines
- (Systems Engineering, Electrical and Computer Engineering, Industrial Engineering)
- Modify/build a software program that introduces students to software development tools /environments
- Troubleshoot and debug changes made to an existing software program
- Develop an original Python software program, learning basic Python language syntax
- Build a foundation for academic success in the Software Engineering degree program.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe various software engineering processes and activities	C1	Knowledge
2. Apply the system modeling techniques to model a medium size software system	C3	Apply
3. Apply software quality assurance and testing principles to medium size software system.	C4	Analyze
4. Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C2	Understanding

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

Course Content:

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

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing.

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Software Engineering, Sommerville I., 11th Edition, Pearson Inc.
2. Farley, D. (2022) Modern Software Engineering: Doing what works to build better software faster. Boston: Addison-Wesley.
3. Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 9th Edition, McGraw-Hill.

Artificial Intelligence

Credit Hours	4 (3-1)	**Prerequisites**	Discrete Structures

Course Introduction:

This course teaches what every student should know about Artificial Intelligence. AI is a fast-moving technology with impacts and implications for both our individual lives and society as a whole. In this course, students will get a basic introduction to the building blocks and components of artificial intelligence, learning about concepts like algorithms, machine learning, and neural networks. Students will also explore how AI is already being used, and evaluate problem areas of AI, such as bias. The course also contains a balanced look at AI's impact on existing jobs, as well as its potential to create new and exciting career fields in the future. Students will leave the course with a solid understanding of what AI is, how it works, areas of caution, and what they can do with the technology.

Course Objectives:

At the end of the course, the students will be able to:

- Build an understanding of the fundamental concepts of artificial intelligence.
- Familiarize the student with the basic taxonomy and terminology of the artificial intelligence.
- Allow the student to gain insight in some specific areas of machine learning and system design.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand key components in the field of artificial intelligence	C2	Understanding
2. Implement classical artificial intelligence techniques	C3	Apply
3. Analyze artificial intelligence techniques for practical problem solving	C4	Analyze

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

Course Content:

Introduction (Introduction, basic component of AI, Identifying AI systems, branches of AI, etc.); Reasoning and Knowledge Representation (Introduction to Reasoning and Knowledge Representation, Propositional Logic, first order Logic); Problem Solving by Searching (Informed searching, Uninformed searching, Local searching.); Constraint Satisfaction Problems; Adversarial Search (Min-max algorithm, Alpha beta pruning, Game-playing); Learning (Unsupervised learning, Supervised learning, Reinforcement learning) ;Uncertainty handling (Uncertainty in AI, Fuzzy logic); Recent trends in AI and applications of AI algorithms (trends, Case study of AI systems, Analysis of AI systems)

Teaching Methodology:

Lectures, Assignments, labs, Projects, Presentations, etc. Major component of the course should be covered using conventional lectures. Practical contact hours are compulsory (~45 hours in a semester).

Course Assessment:
Exams, Assignments, Quizzes, Project, Presentations. Course will be assessed using a combination of written examinations and project(s). Practical evaluation, using rubrics, is encouraged, and suggested to make up around 20% of the course.
Reference Materials:
1. LARSON, E.R.I.K.J. (2022) Myth of Artificial Intelligence: Why computers can't think the way we do. S.l.: THE BELKNAP PRESS. 2. Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 4th edition, Prentice Hall, Inc. 3. Hart, P.E., Stork, D.G. and Duda, R.O., Pattern classification. John Willey & Sons. 4. Luger, G.F. and Stubblefield, W.A. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley.

Digital Logic Design

Credit Hours	4 (3-1)	**Prerequisites**	Applied Physics

Course Introduction:

This is core course that presents basic tools for the design of digital circuits. It serves as a building block in many disciplines that utilize data of digital nature like digital control, data communication, digital computers etc.

Course Objectives:

The objective of this course includes:

- To understanding importance of logic gates.
- To understand concepts and terminologies of digital logic design.
- To understand the operating logic of the gates in combinational and sequential logic circuits.
- To Introduce to application of digital logic.
- To enable student to design digital circuitry, analyze and interpret data

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire knowledge related to the concepts, tools, and techniques for the design of digital electronic circuits	C1	Knowledge
2. Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques	C3	Apply
3. Apply the acquired knowledge to simulate and implement small-scale digital circuits	C4	Analyze
4. Understand the relationship between abstract logic characterizations and practical electrical implementations.	C2	Understanding

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

Course Content:

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

Teaching Methodology:
Lectures, Written Assignments, Practical labs, Semester Project, Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam
Reference Materials:
1. TARAATE, V.A.I.B.B.H.A.V. (2022) Digital design techniques and exercises: A practice book for digital logic design. S.l.: SPRINGER. 2. Digital Fundamentals by Floyd, 11th edition. 3. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 3rd edition.

Design and Analysis of Algorithms

Credit Hours	3 (3-0)	**Prerequisites**	Data Structures and Algorithms

Course Introduction:

This core course covers good principles of algorithm design, elementary analysis of algorithms, and fundamental data structures. The emphasis is on choosing appropriate data structures and designing correct and efficient algorithms to operate on these data structures.

Course Objectives:

The main objectives of this course are to:

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm	C1	Knowledge
2. Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors.	C5	Evaluate
3. Determine informally the time and space complexity of simple algorithms	C4	Analyze
4. List and contrast standard complexity classes	C2	Understanding
5. Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms	C3	Apply
6. Use of the strategies (brute-force, greedy, divide-and-conquer, and dynamic programming) to solve an appropriate problem	C3	Apply
7. Solve problems using graph algorithms, including single source and all-pairs shortest paths, and at least one minimum spanning tree algorithm	C3	Apply
8. Trace and/or implement a string-matching algorithm	C3	Apply

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

Course Content:

Introduction; role of algorithms in computing, Analysis on nature of input and size of input Asymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω, Sorting Algorithm analysis, loop invariants, Recursion and recurrence relations; Algorithm Design Techniques, Brute Force Approach, Divide-and-conquer approach; Merge, Quick Sort, Greedy approach; Dynamic programming; Elements of Dynamic Programming, Search trees; Heaps; Hashing; Graph algorithms, shortest paths, sparse graphs, String matching; Introduction to complexity classes;
Teaching Methodology:
Lectures, Written Assignments, Semester Project.
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Project, Final Exam
Reference Materials:
1. Introduction to Algorithms (3rd edition) by Thomas H. Corman, Charles E. 2. Leiserson, Ronald L. Rivest and Clifford Stein Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos, 3. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne

Computer Organization and Assembly Language

Credit Hours	4 (3-1)	**Prerequisites**	Programming Fundamentals

Course Introduction:

Computer Organization and Assembly Language Programming deals with lower-level computer programming—machine or assembly language, and how these are used in the typical computer system. The book explains the operations of the computer at the machine language level.

Course Objectives:

At the end of the course, the students will be able to:

- Understand the internal working and organization of various building blocks of a digital computer as well as simple assembly language programming techniques.
- Understand the Assembler and Debugger, Manipulate and translate machine and assembly code.
- Describe actions inside the processing chip.

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire the basic knowledge of computer organization, computer architecture and assembly language	C1	Knowledge
2. Understand the concepts of basic computer organization, architecture, and assembly language techniques	C2	Understanding
3. Solve the problems related to computer organization and assembly language	C3	Apply

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

Course Content:

Introduction to computer systems: Information is bits + context, programs are translated by other programs into different forms, it pays to understand how compilation systems work, processors read and interpret instructions

stored in memory, caches matter, storage devices form a hierarchy, the operating system manages the hardware, systems communicate with other systems using networks; Representing and manipulating information: information storage, integer representations, integer arithmetic, floating point; Machine-level representation of programs: a historical perspective, program encodings, data formats, accessing information, arithmetic and logical operations, control, procedures, array allocation and access, heterogeneous data structures, putting it together: understanding pointers, life in the real world: using the gdb debugger, out of-bounds memory references and buffer overflow, x86-64: extending ia32 to 64 bits, machine-level representations of floating-point programs; Processor architecture: the Y86 instruction set architecture, logic design and the Hardware Control Language (HCL), sequential Y86 implementations, general principles of pipelining, pipelined Y86 implementations
Teaching Methodology:
Lectures, Written Assignments, Practical labs, Semester Project, Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam
Reference Materials:
1. Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e), Randal E. Bryant and David R.O' Hallaron, Carnegie Mellon University 2. Robert Britton, MIPS Assembly Language Programming, Latest Edition, 3. Computer System Architecture, M. Morris Mano, Latest Edition, 4. Assembly Language Programming for Intel- Computer, Latest Edition

Parallel and Distributed Computing			
Credit Hours	3 (3-0)	**Prerequisites**	Operating Systems
Course Introduction:			
This course covers general introductory concepts in the design and implementation of parallel and distributed systems, covering all the major branches such as Cloud Computing, Grid Computing, Cluster Computing, Supercomputing, and Many-core Computing. The specific topics that this course will cover are: 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, and synchronization.			
Course Objectives:			
The primary goal of parallel computing is to increase available computation power for faster application processing and problem solving.			
Course Learning Outcomes (CLOs):			
At the end of the course the students will be able to:		**Domain**	**BT Level***

1. Learn about parallel and distributed computers. 2. Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library 3. Analytical modelling and performance of parallel programs. 4. Analyze complex problems with shared memory programming with open MP.	C1 C2 C3 C4	Knowledge Understanding Apply Analyze
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		
Course Content:		
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).		
Teaching Methodology:		
Lectures, Written Assignments, Practical labs, Semester Project, Presentations		
Course Assessment:		
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam		
Reference Materials:		
1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum and M. V. Steen, Prentice Hall, 2nd Edition, 2007 2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.		

Compiler Construction

Credit Hours	3 (3-0)	**Prerequisites**	Theory of Automata

Course Introduction:

Compiler construction is an area of computer science that deals with the theory and practice of developing programming languages and their associated compilers. The theoretical portion is primarily concerned with syntax, grammar, and semantics of programming languages.

Course Objectives:

The main course objectives of this course are to:

- Differentiate between different levels of programming languages.
- Understand the role of front-end and back-end of a compiler. Recognize different types of grammars.
- Understand and define grammars in BNF, syntax diagrams, regular expressions.
- Define tokens using the notation of regular expressions. Convert regular expressions into finite automata. Implement a lexical analyzer. Define a programming language syntax using a CFG.
- Construct a parse tree for a given program. Differentiate between top-down and bottom-up parsing strategies. Understand LL (k) and LR (k) grammars.
- Write a top-down parser using recursive-descent and LL (1) parsing methods. Understand simple-precedence, operator precedence and SLR parsing methods.
- Understand semantic analysis (type checking, scope checking etc.)
- Understand various types of runtime environments. Understand code generation techniques. Understand code optimization techniques.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation	C2	Understanding
2. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines	C3	Apply
3. Design and implement a compiler using a software engineering approach	C6	Create
4. Use generators (e.g., Lex and Yacc)	C3	Apply

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

Course Content:

Introduction to interpreter and compiler. Compiler techniques and methodology; Organization of compilers; Lexical and syntax analysis; Parsing techniques. Types of parsers, top-down parsing, bottom-up parsing, Type checking, Semantic analyzer, Object code generation and optimization, detection and recovery from errors.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

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

Reference Materials:

1. Thain, D. (2020) Introduction to compilers and Language Design. United States?: Douglas Thain.
2. Watson, D. (2017) A practical approach to compiler construction. Cham Switzerland: Springer.
3. Compilers: Principles, Techniques, and Tools, A. V. Aho, R. Sethi and J. D. Ullman, Addison-Wesley, 2nd ed., 2006

Theory of Automata

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

Theory of Automata is an exciting, theoretical branch of computer science. It established its roots during the 20th Century, as mathematicians began developing - both theoretically and literally - machines which imitated certain features of man, completing calculations more quickly and reliably

Course Objectives:

Introduce concepts in automata theory and theory of computation. Identify different formal language classes and their relationships. Design grammars and recognizers for different formal languages. Prove or disprove theorems in automata theory using its properties.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain and manipulate the different concepts in automata theory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc;	C1	Knowledge
2. Prove properties of languages, grammars, and automata with rigorously formal mathematical methods	C2	Understanding
3. Design of automata, RE and CFG	C3	Apply
4. Transform between equivalent NFAs, DFAs and Res	C4	Analyze)
5. Define Turing machines performing simple tasks.	C6	Create
6. Differentiate and manipulate formal descriptions of languages, automata, and grammars with focus on regular and context-free languages, finite automata, and regular expressions.	C3	Apply

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

Course Content:

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.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

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

Reference Materials:

1. Singh, A. (2020) Formal languages and automata theory. S.I.: Amazon LLC, Patna, ACT.
2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011
3. An Introduction to Formal Languages and Automata, by Peter Linz, 4th edition, Jones & Bartlett Publishers, 2006
4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, 2005, New Age Publishers

Multi-variate Calculus

Credit Hours	3 (3-0)	**Prerequisites**	Calculus & Analytical Geometry

Course Introduction:

In Mathematics, multivariable calculus is also known as multivariate calculus. Multivariable calculus is the study of calculus in one variable to functions of multiple variables. The differentiation and integration of multivariable calculus include two or more variables, rather than a single variable.

Course Objectives:

The course develops students' fundamental skills of solving ordinary differential equations and developing differential equations for real-world problems.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basic concepts and know the basic techniques of differential and integral calculus of functions of several variables.	C2	Understanding
2. Apply the theory to calculate the gradients, directional derivatives, arc length of curves, area of surfaces, and volume of solids	C3	Apply
3. Solve problems involving maxima and minima, line integral and surface integral, and vector calculus	C3	Apply

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

Course Content:

Functions of Several Variables and Partial Differentiation; Multiple Integrals: Line and Surface Integrals; Green's and Stoke's Theorem; Fourier Series; periodic functions: Functions of any period P-2L: Even & odd functions: Half Range expansions; Fourier Transform; Laplace Transform; Z-Transform.

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. SHEA., L.A.X.P.E.T.E.R.D.T.E.R.R.E.L.L.M.A.R.I.A. (2018) Multivariable calculus with applications. SPRINGER NATURE.
2. Stewart, J., Clegg, D. and Watson, S. (2021) Multivariable calculus. Australia: Cengage.
3. Multivariable Calculus, 6th edition James, Stewart 2007 Cengage Learning publishers.

4. Calculus and Analytical Geometry, 6th edition. Swokowski, Olinick and Pence. 1994.Thomson Learning EMEA, Ltd.
5. Multivariable Calculus, 5th edition Howard, A. Albert, H. 1995, John Wiley.

Graph Theory

Credit Hours	3 (3-0)	**Prerequisites**	None

Course Introduction:

In the domain of mathematics and computer science, graph theory is the study of graphs that concerns with the relationship among edges and vertices.

Course Objectives:

- Introduce the fundamental concepts of Graph Theory
- Be able to describe the design issues relating to the architectural options
- Provide knowledge for application of Graph Theory in subsequent courses

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Introduce the fundamental concepts of Graph Theory. 2. Provide knowledge for application of Graph Theory in subsequent courses in the design and analysis of algorithms, computability theory, software engineering, and computer systems.	C1 C2	Knowledge Understanding

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

Course Content:

Introduction to Graph Theory; Basic definitions: computer representations and properties of Graph: Data structure for representing Graphs; Fundamental theorem of Graph Theory; Isomorphic and Special Graphs: Properties of Trees and Forests; Binary tree: Balanced binary tree: Directed and Undirected rooted tree: Minimum Spanning Tree algorithms and implementation; Path and Distance in graphs; Shortest path algorithms and implementation; Cycle and distance in weighted graph and digraphs; Distance algorithms and implementation; Eulerian graphs and Hamiltonians graphs with applications; Flow networks: Max-flow Min-cut Theorem; Graph coloring: Edge coloring; Planar graphs; Four color theorem; Deadlock of computer system; Matching Algorithms; Dominance & Ramsey theory

Teaching Methodology:

Lecturing, Written Assignments, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Graph Theory & Applications (1st Edition) by Fournier. Published by Wiley-ISTE, 2011.
2. Applied Algorithmic Graph Theory (1st Edition) by Chartrand. Published by McGraw-Hill College, 1995.

3. Handbook of Graph Theory (Series Edition) by Jonathan Published by CRC Press, 2004.

Theory of Programming Languages

Credit Hours	3 (3-0)	**Prerequisites**	Programming Fundamentals

Course Introduction:

A theoretical study of programming languages. Introduction to grammars and parsers. Language design issues and practical applications

Course Objectives:

1. Familiarize students with basic theory of programming languages
2. Enable to design semantic model of a programming language

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. The better understating the underlying theory of programming languages	C1	Knowledge
2. Enable a student to choose the appropriate Language for a Project.	C2	Understanding
3. Learning of formal semantics design for a programming Languages.	C2	Understanding

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

Course Content:

Introduction: Models of Computation, Syntax and Semantics, Pragmatics, Language Design Principles. Syntax and Semantics: Context-Free Grammars, Regular Expressions, Attribute Grammars and Static Semantics, Algebraic Semantics, Axiomatic Semantics, Denotational Semantics. BNF grammars and Syntax, Operational Equivalence, Abstraction and Generalization, Expressions, Assignment Statement, and Control Structures, Functional Programming: The Lambda Calculus, Operational Semantics, Reduction Order, Recursive Functions, Logic Programming, Inference Engine, Concurrency.

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Concepts of Programming Languages, Robert W. Sebesta, 10th edition, 2012
2. Scott, Michael L., Programming Language Pragmatics, 2nd edition, 2006
3. Theory Introduction to Programming Languages, by Anthony A. Aaby, 2004

Numerical Computing

Credit Hours	3 (3-0)	**Prerequisites**	Calculus and Analytical Geometry

Course Introduction:

Numerical computing is an approach for solving complex mathematical problems using only simple arithmetic operations

Course Objectives:

1. On completion of this course, students will be able to demonstrate programming proficiency using structured programming techniques to implement numerical methods for solutions using computer-based programming techniques using MATLAB for all methods.
2. The course must serve the purpose of scientific software development for science and engineering problems

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. The student would understand the fundamental concepts of Scientific Programming using programing Language(s)	C1	Knowledge
2. Use a computer algebra system to investigate and solve mathematical problems relating to integration, differential equations, and approximation.	C2	Understanding

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

Course Content:

Mathematical preliminaries and error analysis, round-off errors and computer arithmetic, Calculate Divided Differences. Use Divided-difference Table. Find Newton's Interpolation Polynomial. Calculate Interpolation with Equally Spaced Data. Find the Difference Table. Calculate, Newton's Forward & Backward Difference Formulae. Use Gauss Formulae. Use Stirling's Interpolation Formula. Use Bessel's Interpolation Formula. Use Everett's Interpolation Formula. Solve Nonlinear Equations. Solve Equations by Bisection Method. Solve Equations by Regula Falsi Method. Solve Equations by Secant Method. Solve Equations by Newton-Raphson Method. Find Fixed Point Iteration. Solve Equations by Jacobi Iterative Methods. Solve Equations by Gauss Seidel Method Calculate Numerical Differentiation. Find Numerical Differentiation Formulae Based on Equally Spaced Data. Find Numerical Differentiation Based on Newton's Forward Differences. Find Numerical Differentiation Based on Newton's Backward Differences. Find Numerical Differentiation Based on Stirling's Formula. Find Numerical Differentiation Based on Bessel's Formula. Find Numerical Differentiation Based on Lagrange's Formula. Calculate Error Analysis of Differentiation Formulae. Solve Richardson Extrapolation. Calculate Numerical Integration. Use Trapezoidal Rule with Error Term. Use Simpson's 1/3 Rule with Error Term. Use Simpson's 3/8 Rule with Error Term. Use Composite Numerical Integration. Use Composite Trapezoidal Rule. Use Composite Simpson's Rule. Find Richardson's Extrapolation. Find Newton-Cotes Closed Quadrature Formulae.

Teaching Methodology:

Lecturing, Written Assignments, Project, Practical Labs, Lab Assignments, Presentations

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Driscoll, T.A. and Braun, R.J. (2018) Fundamentals of Numerical Computation. Philadelphia: Society for Industrial and Applied Mathematics.

2. Numerical Analysis (9th Edition) by Richard L. Burden, J. Douglas Faires by Brooks/Cole Boston USA, 2011
3. Numerical Methods for Scientific Computing by J.H. Heinbockel Trafford Publishing USA, 2006

Differential Equations

Credit Hours	3 (3-0)	**Prerequisites**	Calculus and Analytical Geometry

Course Introduction:

A description of how something continuously changes over time. Some differential equations can have an analytical solution such that all future states can be known without simulation of the time evolution of the system. However, most can have a numerical solution with only limited accuracy.

Course Objectives:

1. The course develops students' fundamental skills of solving ordinary differential equations, and developing differential equations for real-world problems

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Identify, analyze, and subsequently solve physical situations whose behavior can be described by ordinary differential equations.	C 2,3	Understanding & Apply
2. Determine solutions to first order separable differential equations	C2	Understanding
3. Determine solutions to first order linear differential equations.	C2	Understanding
4. Determine solutions to first order linear differential equations.	C2	Understanding
5. Determine solutions to first order linear differential equations.	C2	Understanding

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

Course Content:

Ordinary differential equations of the first order; Geometrical considerations; Isoclines; Separable equations; Equations reducible to separable form; Exact differential equations; Integrating factors; Linear first-order differential equations; Variation of parameters; Ordinary linear differential equations; Homogeneous linear equations of the second order; Homogeneous second order equations with constant coefficients; General solution; Real roots; Complex roots; Double root of the characteristic equation; Differential operators; Cauchy equation; Homogeneous linear equations of arbitrary order; Homogeneous linear equations of arbitrary order with constant coefficients; Non-homogeneous linear equations; Modeling of electrical circuits; Systems of differential equations; Series solutions of differential equations; Partial differential equations; Method of separation of variables; Laplace equations and their solutions by Fourier series method.

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Bronson, R. and Costa, G.B. (2022) Differential equations. New York: McGraw-Hill 2. A First Course in Differential Equation Zill. Prindle. Weber. Schmidt.1996. Brooks/Cole Publishing. 3. Differential Equations with Boundary-Value Problems, Dennis. G. Zill, Michael, R. Cullen. 1996, Brooks/Cole Publishing, 4. Elementary Differential Equations with Applications C. H. Edwards. David, E. 1993. Penney, Prentice Hall.

Advance Database Systems			
Credit Hours	3 (3-0)	**Prerequisites**	Database Systems

Course Introduction:

This course focuses on research and applications in advanced database systems for Cloud and Big Data Computing. It provides an opportunity to learn about Cloud Computing and Advanced Database Systems and apply that learning on a popular cloud platform. The course topics include how database systems have addressed the four Vs of Big Data: volume, variety, velocity, and veracity. We also consider maintaining the virtue of our data, a fifth V if you will, by addressing issues of security, privacy, and social responsibility.

Course Objectives:

The course objectives are the following:

- To provide the students with a better understanding of the essential techniques used in a Database Management System, either by revisiting them or by studying new approaches.
- To provide students with knowledge to choose, design, and implement a database management system in a complex domain, making the best use of the available tools and techniques.
- To provide students with knowledge to analyze and tune a given database management system, given a workload and usage patterns.
- To allow the students to learn and experiment advanced database techniques, models, and products, and to provide them with the knowledge to take decisions concerning implementation issues.
- To provide students with knowledge to analyze, modify if necessary and experiment algorithms that make up the database internals.
- To expose students to advanced topics and techniques that appear promising research directions.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe database management system internals. Understand and describe internal algorithms in detail.	C2	Understanding
2. Identify and be able to use recent and advanced database techniques.	C1	Knowledge
3. Decide on configuration issues related to database operation and performance. Identify which parameters are tunable and what are the implications.	C6	Decision-making
4. Analyze, describe, and use other models than the Relational.	C3	Analysis

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

Course Content:

Database systems concepts and architecture. Concepts used in UML, EER, and XML. Transformation of conceptual models to a relation. Properties of normalization up to 4NF. Views, implementation of integrity constraints. Centralized, decentralized, and distributed databases. Transaction handling. Concurrency and recovery. Query optimization. Advanced and embedded SQL. Triggers and stored procedures. The problem of using different architectures in client and server-side applications. Techniques for efficient storing, accessing, securing, and recovering of data. Implementation of advanced structures in relational, hybrid, and object-oriented databases. Techniques for distributed databases.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Carpenter, J. & Hewitt, E. (2022). Cassandra: the definitive guide (2nd ed.). O'Reilly Media, Inc. The second edition is available used or in overstock at a much lower price from the third edition. The second edition is sufficient for our needs. 2. Damji, J., Lee, D., Wenig, B., & Das, T. (2020). Learning Spark: lightning-fast big data analysis (2nd ed.) O'Reilly Media, Inc. 3. Harrison, G. (2016). Next generation databases: NoSQL, newSQL, and big data. Apres. Look for it used or in overstock on the Internet for a much lower price. 4. Perkins, L., Redmond, E., & Wilson, J. (2018). Seven databases in seven weeks: a guide to modern databases and the NoSQL movement. Pragmatic Bookshelf.

Machine Learning

Credit Hours	3 (2-1)	**Prerequisites**	Programming for Artificial Intelligence

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:

- Present the basic machine learning concepts.
- Present a range of machine learning algorithms along with their strengths and weaknesses.
- Apply machine learning algorithms to solve problems of moderate complexity.

Course Objectives:

The main objective of this course is to make students comfortable with tools and techniques required in handling large amounts of datasets. They will also uncover various deep learning methods in NLP, Neural Networks etc.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe basic machine learning concepts, theories, and applications.	C1	Knowledge
2. Apply supervised learning techniques to solve classification problems of moderate complexity.	C3	Apply
3. Apply unsupervised learning techniques to solve clustering problems of moderate complexity	C3	Apply
4. Apply reinforcement learning algorithms to environments with complex dynamics.	C3	Apply
5. Develop a reasonable size project using suitable machine learning technique	C6	Create

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

Course Content:

Introduction to machine learning; concept learning: General-to-specific ordering of hypotheses, Version spaces Algorithm, Candidate elimination algorithm; Supervised Learning: decision trees, Naive Bayes, Artificial Neural Networks, Support Vector Machines, Overfitting, noisy data, and pruning, Measuring Classifier Accuracy; Linear and Logistic regression; Unsupervised Learning: Hierarchical Agglomerative Clustering. k-means partitional clustering; Self-Organizing Maps (SOM) k-Nearest-neighbor algorithm; Semi-supervised learning with EM using labeled and un-labeled data; Reinforcement Learning: Hidden Markov models, Monte Carlo inference Exploration vs. Exploitation Trade-off, Markov Decision Processes; Ensemble Learning: Using committees of multiple hypotheses. Bagging, boosting.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Machine Learning, Tom, M., McGraw Hill, 1997.
2. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, MIT Press,2012

Artificial Neural Networks

Credit Hours	3 (2-1)	**Prerequisites**	Programming for Artificial Intelligence

Course Introduction:

This course will introduce Artificial Neural Networks, their basic architecture and how they mimic the human brain using simple mathematical models. Many of the important concepts and techniques around brain computing and the major types of ANN will also be introduced. Emphasis is made on the mathematical models, understanding learning laws, selecting activation functions and how to train the networks to solve classification problems. Students would be able to understand and use different types of neural networks and would be able to use different activation functions and construct layered networks to solve classification problems.

Course Objectives:

The objective of this course is to trace the historical developments of artificial intelligence leading to artificial neural networks (ANN). The course introduces the basic concepts and models of ANN for solving simple pattern recognition problems.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the fundamentals of neural networks in AI	C2	Understanding
2. Explain how simple ANNs can be designed	C2	Understanding
3. Apply ANN for classification Problems	C3	Apply
4. Differentiate between different Networks and their learning laws	C4	Analyze

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

Course Content:

Introduction and history of neural networks, Basic architecture of neural networks, Perceptron and Adaline (Minimum Error Learning) for classification, Gradient descent (Delta) rule, Hebbian, Neo-Hebbian and Differential Hebbian Learning, Drive Reinforcement Theory, Kohonen Self Organizing Maps, Associative memory, Bi-directional associative memory (BAM), Energy surfaces, The Boltzmann machines, Backpropagation Networks, Feedforward Networks; Introduction to Deep learning and its architecture.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Aggarwal, C.C. (2019). Neural networks and deep learning: A textbook. New York: Springer.
2. Neural Network Design, 2nd Edition, Martin T. Hagan, Howard, B. Demuth, Mark
3. Hudson Beale and Orlando De Jesus, Publisher: Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 0971732116
4. Fundamentals of Artificial Neural Networks, Mohammad Hassoun, Publisher: A Bradford Book (January 1, 2003), ISBN-10: 0262514672

Cloud Computing

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

Course Introduction:

The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its focus is on parallel programming techniques for cloud computing and large-scale distributed systems, which form the cloud infrastructure. The topics include overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art solutions for cloud computing developed by Google, Amazon, Microsoft, Yahoo, VMWare, etc. Students will also apply what they learn in one programming assignment and one project executed over Amazon Web Services.

Course Objectives:

Upon successful completion of this course, students will learn:

- The fundamental ideas behind Cloud Computing, the evolution of the paradigm, its applicability benefits, as well as current and future challenges.
- The basic ideas and principles in data center design; cloud management techniques and cloud software deployment considerations.
- Different CPU, memory and I/O virtualization techniques that serve in offering software, computation, and storage services on the cloud; Software Defined Networks (SDN) and Software Defined Storage (SDS).
- Cloud storage technologies and relevant distributed file systems, NoSQL databases and object storage.
- The variety of programming models and develop working experience in several of them.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.	C3	Analyze
2. Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google App-Engine.	C4	Apply
3. Solve a real-world problem using cloud computing through group collaboration.	C2	Problem Solving
4. Identify security and privacy issues in cloud computing.	C1	Knowledge

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

Course Content:

Definition and evolution of Cloud Computing. Enabling Technologies, Service and Deployment Models Popular Cloud Stacks and Use Cases. Benefits, Risks, and Challenges of Cloud Computing. Economic Models and SLAs. Topics in Cloud Security. Historical Perspective of Data Centers. Datacenter Components: IT Equipment and Facilities. Design Considerations: Requirements, Power, Efficiency, & Redundancy. Power Calculations, PUE and Challenges in Cloud Data Centers. Cloud Management and Cloud Software Deployment Considerations. Virtualization (CPU, Memory, I/O). Case Study: Amazon EC2. Software Defined Networks (SDN). Software Defined Storage (SDS). Introduction to Storage Systems. Cloud Storage Concepts. Distributed File Systems (HDFS, Ceph FS). Cloud Databases (HBase, MongoDB, Cassandra, DynamoDB). Cloud Object Storage (Amazon S3, OpenStack Swift, Ceph). Distributed Programming for the Cloud. Data-Parallel Analytics with Hadoop MapReduce (YARN). Iterative Data-Parallel Analytics with Apache Spark. Graph-Parallel Analytics with GraphLab 2.0 (PowerGraph)

Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Marinescu, Dan (2017) Cloud Computing Theory and Practice (2nd Ed.) 2. IEEE Transactions on Cloud Computing 3. Journal of Cloud Computing: Advances, Systems and Applications (JoCCASA)

Text Mining			
Credit Hours	3 (3-0)	**Prerequisites**	Data Structure

Course Introduction:

Given the dominance of text information over the Internet, mining high-quality information from text becomes increasingly critical. The actionable knowledge extracted from text data facilitates our life in a broad spectrum of areas, including business intelligence, information acquisition, social behavior analysis and decision-making. In this course, we will cover important topics intext mining including basic natural language processing techniques, document representation, text categorization and clustering, document summarization, sentiment analysis, social network and social media analysis, probabilistic topic models and text visualization.

In addition, as we are in the era of Big Data, we will provide you opportunities to gain hands-on experience of handling large-scale data set, i.e., Big Data.

Course Objectives:

Upon successful completion of this course, students will be able to:

- Develop key text and data mining knowledge and understanding through presentations, hands-on coding lessons and the production of research material via their project.
- Practice the use of computational methods to analyze text collections as a technique to answer scholarly research questions.
- gain autonomy, accountability and learn to work with others by collaborating in small groups on the practical elements of the course and during the preparation stage of their project, developing their communication skills, and gaining valuable skills in working with others.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain and use text preprocessing techniques	C2	Understanding
2. Describe a text analytics system together with its components, optional and mandatory ones	C1	Knowledge
3. Explain how text could be analyzed	C2	Understanding
4. Evaluate results of text analytics	C3	Problem Solving
5. Analyze and reflect on the various techniques used in text analytics and the parameters needed as well as the problem solved	C3	Analysis
6. Plan & execute a text analytics experiment	C4	Create

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

Course Content:
The Text Mining course is focusing on the importance and the difficulty of analyzing text. The Text Mining course is designed to provide students with knowledge relevant to both preprocessing of text as well as analytics of text. The Text Mining course, however, focuses on wide range of algorithms, techniques, and tools. These include standard methods, such as: tokenization, TF-IDF, n-grams, Named Entity Extraction, Sentiment Analysis, and Topic Modeling. Furthermore, recent trends in machine learning and deep learning are also covered, including: Word2Vec, Semantic Hashing, and Recurrent Neural Networks for Natural Language Processing. Various examples and use cases are used across the course.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Aggarwal, C.H.A.R.U.C. (2019) Machine Learning For Text. S.L.: Springer. 2. Lamba, M. and Madhusudhan, M. (2022) Text mining for information professionals an uncharted territory. Cham, Switzerland: Springer. 3. Miner, G. (2016) Practical text mining and statistical analysis for non-structured text data applications. Amsterdam: Academic Press. 4. Mining Text Data. Charu C. Aggarwal and ChengXiang Zhai, Springer, 2012.

Fundamentals of Internet of Things (IoT)			
Credit Hours	3 (3-0)	**Prerequisites**	
Course Introduction:			
Internet of Things (IoT) course is all about understanding and then developing solid skills to build IoT systems. This is a very hands-on intensive and interactive course. Much of the course material will be delivered in using the flipped lectures-based model where pre-work will be given to students before they come to sessions. Largely, the sessions are based on hands-on workshops where students will perform different programming and development tasks.			
Course Objectives:			
Upon successful of completion of this course, students will be able to: • Use the FIT IOT-LAB for development of testbeds for network computer communications. • IBM BLUEMIX for Cloud Development. • Arduino and Raspberry Pi for building embedded systems. • TelosB Motes. • SDN based IoT.			
Course Learning Outcomes (CLOs):			
At the end of the course the students will be able to:		**Domain**	**BT Level***
1. Explain and define Internet of Things in different contexts. 2. Take account of the key components that make up an IoT system.		C2 C1	Understanding Knowledge

3. Differentiate between the levels of the IoT stack and be familiar with the key technologies and protocols employed at each layer of the stack.	C3	Analyze
4. Apply the knowledge and skills acquired during the course to build and test a complete, working IoT system involving prototyping and programming.	C4	Apply
5. Understand where the IoT concept rightly fits within the broader ICT industry and possible future trends.	C2	Understanding

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

Course Content:

ADCs/DACs, PVM and Voltage Dividers. Understanding the OSI model and the seven abstraction layers. Networking and TCP/IP. Control and Management plane improvements with SDN. Openness. Network Automation and Virtualization. SDN and OpenStack. ONOS SDN Controllers. Applications and APIs. Protocols. Arduino and Raspberry Pi Programming. Elements of an IoT ecosystem. Technology and business drivers. IoT applications, trends, and implications. Sensing components and devices. Sensor modules, nodes, motes, and systems. Wireless technologies for the IoT. Edge connectivity and protocols. Wireless sensor networks. Local processing on the sensor nodes. Connecting devices at the edge and to the cloud. Processing data offline and in the cloud.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. J. Biron and J. Follett, "Foundational Elements of an IoT Solution", O'Reilly Media, 2016.
2. Keysight Technologies, The Internet of Things: Enabling Technologies and Solutions for Design and Test Application Note, 2016.

Human Computer Interaction

Credit Hours	3 (3-0)	**Prerequisites**	Software Engineering

Course Introduction:

This course studies how best to design the interface between human users and computer systems. Emphasis is placed on learning how to involve the user at different stages in the design process to improve the interface in a cost-effective way. In particular, experience with iterative user-centered design, rapid prototyping and usability testing methods are developed. Students evaluate several computer interfaces as well as iteratively design and evaluate an interface prototype.

Course Objectives:

- Course introduces the main concepts of designing, evaluating and functional deploying, effectual technologies in a range of circumstance - be it office, home, school, internet world or another domain.
- The objective of this course is to give an introduction to the key areas, accessing and design developments in the field. The course aims, understanding and importance of UI its design and mistakes.
- The course helps to learn basics concepts of field such as, design rules and guidelines, prototyping and design patterns for interactive systems

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain context of HCI and different measures for evaluation.	C2	Understanding
2. Apply the principles of good design for people from the perspective of age and disabilities.	C3	Apply
3. Analyze techniques for user centered design for a medium sized software.	C4	Analyze
4. Evaluate the usability of a medium size software user interface.	C5	Evaluate

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

Course Content:

Contexts for HCI, Psychology of usable things, Processes for User-Centered Design, Metrics and Measures for Evaluation, Usability heuristics and principles of Usability testing, Physical capabilities, Cognitive and social models for interaction design, Principles of good interaction design, Accessibility, Principles of GUI, Visual design elements, Data gathering, Task analysis, Prototyping, Help and user documentation, Internationalization, Usability inspection methods, Usability testing methods, New Interaction Technologies, Usability in practice, Visual Design and Typography, Icon Design, Ubiquitous, Augmented and Virtual Reality.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Designing the User Interface: Strategies for Effective Human-Computer Interaction, Ben Shneiderman and Catherine Plaisant, 6th Ed, Pearson Inc, 2016.
2. Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, Benyon, D. 3rd Ed., Pearson. 2013
3. About Face: The Essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, 4th Ed, Wiley, 2014

Real-Time Systems

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

Course Introduction:

This course covers the principles of real-time systems, Modeling of a Real-Time System, Task assignment and scheduling, Resource management, Real-time operating systems, RTOS services, Programming language with real-time support, System design techniques, Inter task communication, Fault tolerant techniques, Reliability evaluation methods; Performance analysis, Case studies of real-time systems.

Course Objectives:

The objective of this course is to

- Develop an understanding of various Real Time systems Application
- Obtain a broad understanding of the technologies and applications for the emerging and exciting domain of real-time systems
- Get in-depth hands-on experience in designing and developing a real operational system.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain fundamental principles for programming of real time systems with time and resource limitations.	C2	Understanding
2. Describe the foundation for programming languages developed for real time programming.	C1	Knowledge
3. Use real time system programming languages and real time operating systems for real time applications.	C4	Apply
4. Analyze real time systems with regard to keeping time and resource restrictions.	C3	Analyze

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

Course Content:

Introduction to task scheduling. Issues in Real Time Computing. Structure of a Real Time System, Task classes. Performance Measures for Real Time Systems. Task Assignment and Scheduling. Classical uniprocessor scheduling algorithms. RM algorithm with different cases. Priority ceiling. Precedence constraints- using of primary and alternative tasks. Uniprocessor scheduling of IRIS tasks. Task assignment. Utilization balancing. Next fit. Bin packing Algorithm. Myopic off-line algorithm Focused addressing and bidding, Buddy strategy, Fault Tolerant Scheduling. Aperiodic scheduling. Spring algorithm, Horn algorithm Bratley Sporadic scheduling. Introduction to Real Time Communication VTCSMA. Case Study-Air traffic controller system. Air traffic controller system. Case Study -Distributed air defense system. Distributed air defense system. Real-time modeling-Introduction. Petri nets and applications in real-time modeling. Applications in real-time modeling. Case Study-Air traffic controller system. Two-phase Approach to improve Predictability.

Maintaining Serialization Consistency. Maintaining Serialization Consistency. Databases for Hard Real Time System. Main Memory Databases Transaction Priorities Transaction Aborts Concurrency control issues. Disk Scheduling Algorithms. Disk Scheduling Algorithms.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Cooling, J.E. (2019) The complete edition - software engineering for real-time systems: A software engineering perspective toward designing real-time systems. Birmingham, UK: Packt Publishing. 2. Shirvaikar, M.U.K.U.L. (2017) Real Time Systems. Cognella Academic Publish. 3. Jermann Kopetz, Real-Time Systems Design Principles for Distributed Embedded Applications, Springer Verlag, 2011. 4. Benjamin M. Brosgol, A Comparison of the Concurrency Features of Ada 95 and Java. 5. The Real-time for Java Expert Group, The Real-Time Specification for JavaTM. 6. Greg Bollella and James Gosling, The Real-Time Specification for Java (summary).

Computer Architecture

Credit Hours	3 (3-0)	**Prerequisites**	Digital Logic Design

Course Introduction:

Modern computer technology requires an understanding of both hardware and software since the interaction between the two offers a framework for mastering the fundamentals of computing. The purpose of this course is to cultivate an understanding of modern computing technology through an in-depth study of the interface between hardware and software. In this course, you will study the history of modern computing technology before learning about modern computer architecture and a number of its essential features, including instruction sets, processor arithmetic and control, the Von Neumann architecture, pipelining, memory management, storage, and other input/output topics. The course will conclude with a look at the recent switch from sequential processing to parallel processing by looking at the parallel computing models and their programming implications.

Course Objectives:

Students who successfully fulfills the course requirements will have demonstrated:

- An ability to understand the design of a pipelined CPU and cache hierarchy
- An ability to analyze and evaluate CPU and memory hierarchy performance
- An understanding of trade-offs in modern CPU design including issues affecting superscalar and dynamically scheduled architectures
- An understanding of hardware design of multiprocessors including cache coherence and synchronization
- Experience with a complex simulation tool to study various micro architectural features.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand functionality of major components of a computer system like CPU, control unit, memory, I/O and storage.	C2	Understanding
2. Understand principles of instruction set design including RISC architectures and basic assembly programming	C2	Problem Analysis
3. Understand pipelining and parallelism features applied in single processor, multiple processors, and multicore architectures	C3	Analyze

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

Course Content:

Introduction, Computer Evolution and Performance. A Top-Level View of Computer Function and Interconnection. Internal Memory Technology. Cache Memory. External Memory. Instruction Sets: Characteristics and Functions. Intel instruction set, Addressing Modes and Formats. Performance measurement and Benchmarking. Input Output Devices. Reduced Instruction Set Computers (RISC). Vs CISC etc. CPU Performance Improvement techniques. Functional parallelism i.e. Pipelining support. Instruction-Level Parallelism and Superscalar Processors. Hardwired / Micro programmed Architectures. Multiprocessor Systems SMP verses Clusters. Multicore Architectures and future directions.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Walter A. Triebel, Avtar Singh, The 8088 and 8086 Microprocessors fourth edition Prentice Hall Inc. 2. William Stallings, Computer Organization and Architecture Designing for performance, 8th Edition, Prentice Hall Inc. 2016 (Textbook)

Systems and Network Administration			
Credit Hours	3 (2-1)	**Prerequisites**	Computer Networks

Course Introduction:

This course introduces students to the foundational concepts and experience in networking and systems administration. The course provides the basic theory, concepts and practical experience in the design, installation and configuration of personal computers, peer-to-peer networks and client-server networks meeting user requirements.

Course Objectives:

At the end of the course the students will be able to:

- Describe the key terminologies and technologies of System and network administration.
- Explain the services and functions provided by different data centers.
- Identify various OS used to handle system and network administration routines.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Design and configure peer-to-peer networks to share resources.	C4	Create
2. Analyze requirements and design network architecture for a given scenario.	C3	Analyze
3. Design and configure IP addressing schemes for given scenario.	C5	Apply
4. Design and configure a client-server network and required network services for a given scenario.	C5	Apply
5. Evaluate and critique a design for a systems and network solution.	C2	Understanding

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

Course Content:

Intro to Networks, OSI interconnect model, topologies, Internet history and TCP/IP. Physical Layer: transmission media, socket programming, UNIX Process Creation and UNIX IPC. Guest Lecturer. Systems Administration topics. Data Link Layer: framing, flow control, error control, encoding for local and wide areas, Admin tricks with UNIX shell. Medium Access Layer. Broadcast, CSMA/CD, CDMA, FDDI, 802.X, Bluetooth. Network Layer: Flow control, congestion control, Routing, quality of service, switching, CIDRs, mobile IP, WAP. Finish Routing. Transport Layer: TCP, UDP, IP v 6. CISCO Router IOS. Application Layer: HTTP, SMTP, DNS, SNMP, FPT, Telnet, streaming video, video compression, multicast, JME. Network Services. Distributed Computing, Network Management. Dist. File Systems.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Study guide for Practice of System and Network Administration by Thomas A. Limoncelli, Cram101; 2nd Edition (2011). ISBN-10: 1428851755. 2. Linux Administration: A Beginner's Guide, Seventh Edition 7th Edition by Wale Soyinka 3. Active Directory: Designing, Deploying, and Running Active Directory Fifth Edition by Barian Desmond

Computer Graphics

Credit Hours	3 (2-1)	**Prerequisites**	

Course Introduction:

An introduction to the computer as a graphic design and artist's tool. Using Macintosh OS, students learn basic use and application of vector illustration (Adobe Illustrator), raster image (Adobe Photoshop), and scanning software programs to the art and design process. Emphasis is on "hands on" use of the computer, and how the computer can aid the artist's and designer's problem-solving process through interactive visual alternatives

Course Objectives:

Upon completing requirements for this course, the student will be able to:

- Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
- Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
- Extract scene with different clipping methods and its transformation to graphics display device.
- Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
- Render projected objects to naturalize the scene in 2D view and use of illumination models for this.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire basic skills in the use of the tools and techniques available in a vector drawing software program.	C1	Knowledge
2. Demonstrate an understanding of the color models used in the creation of digital art and design.	C3	Apply
3. Be able to scan, and import line art, grey scale, and color images.		
4. Understand and choose file formats appropriate for digitally created art and design.	C2	Understand
5. Demonstrate the development of visual and conceptual skills required to create a successful design solution through the process of idea development, refinement, and assessment in the creation of design projects.	C4	Create

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

Course Content:
Vector drawing. Image manipulation. Paint. Image scanning. Page layout. 2/3D animation. 3D modeling. Presentation graphics. Interactive media. Web design. Storage mediums. Network servers and navigation. Personal file system. Backup and master file strategy. Starting up and shutting down. Desktop/finder. Icons. Menus and windows. Dialog boxes. Opening and closing software applications. Creating a file. Saving/copying files I. File management. Printing. Inserting and ejecting media. Shutdown Traditional vs. computer graphics techniques. Choosing the right software program. Quality/resolution. Vector vs. raster. Black and white. Color. Inconsistencies between screen image and final output. Overview of the use of vector graphics. Introduction of Illustrator tools and menu options. Tutorials of Illustrator techniques. Application of learned techniques in creative project. RGB/HLS/CYMK. Pantone or equivalent. Evaluating images. Determining scanning resolution. Line art/grayscale/color. File size and storage issues. Other sources for image acquisition. Copyright issues. Overview of the use of raster images. Introduction of Photoshop tools and menu options. Tutorials of Photoshop techniques. Application of learned techniques in creative project.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Visual Quick Start Guide—Illustrator. PeachPit Press 2. Visual Quick Start Guide—Photoshop. PeachPit Press

Biometric Systems

Credit Hours	3 (3-0)	**Prerequisites**	Programming Fundamentals

Course Introduction:

This is an introductory level course for undergraduate students or practitioners to gain knowledge and hand-on experiences in biometric systems and security applications. Topics include Introduction to important biometric security technologies and policies, biometric modalities and signal processing, biometric solutions and applications, biometric encryption and cryptosystems, biometrics identity analysis and privacy considerations.

Course Objectives:

The student will be able to

- Knowledge of biometric foundations.
- Understanding of behavioral and physical biometric modalities.
- Knowledge of data acquisition techniques on mobile platforms.
- Have awareness of spoofing and common anti-spoofing techniques.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the technological uplifts with biometrics compared to traditional securing mechanisms.	C2	Understanding
2. Gain knowledge in building blocks of research fields like Pattern Recognition, Image Processing and Machine Learning etc.	C1	Knowledge
3. Evaluate and Design security systems with biometrics.	C3	Analyze

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

Course Content:

Introduction to biometric systems. Performance evaluation. Reliability of recognition results. Face detection. Face recognition 2D and 3D. Face spoofing. Ear recognition. Iris recognition. Basics on fingerprints recognition. Other biometrics. Multibiometric systems.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. A.K. Jain, P. Flynn, A.A. Ross, Handbook of Biometrics, Springer, 2008.
2. H. Wechsler, Reliable Face Recognition Methods: System Design, Implementation and Evaluation, Springer, 2007.
3. A.Ross, K. Nandakumar; A.K. Jain. Handbook of Multibiometric. Springer, 2006

Mobile Application Development

Credit Hours	3 (3-0)	**Prerequisites**	Object Oriented Programming

Course Introduction:

Mobile Application Development is market-oriented course in the undergraduate programs of Department of Computer Science at Capital University of Science and Technology. Today, mobile applications are used not only as a standalone application but also with most of web or desktop applications. These applications are highly user focused and designed for every walk of life. Moreover, with the growing strength and cheap availability of mobile devices it has emerged as an important tool in both local and international job market. The course is designed to impart both conceptual and practical knowledge, which is accompanied with hands-on training primarily focused on Android OS, Apple iOS, and related tools. The course demonstrates standard practices and tools used in market to develop robust mobile applications.

Course Objectives:

Upon completing requirements for this course, the student will be able to:

- Create a mobile application using the Swift programming language.
- Debug a mobile application written in the Swift programming language.
- Test a mobile application written in the Swift programming language.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe Mobile Application Development fundamentals and flow on multiple devices and publishing it online	C1	Knowledge
2. Produce Mobile Application using provided assets with basic functionality	C5	Create
3. Make Mobile application that uses hardware and software resources like sensors and configuration etc. and evaluate functionality	C5	Create

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

Course Content:

• Introduction to the course and course objectives. Setting up environment. App Fundamentals. Components of an Application. Introduction to Android and iOS Platform. Developing single screen layout apps. Traversing in screens and data transfers. Storage persistence. Multithreading. Background Services. Notifications services. Testing Applications for data persistence. Exporting installable app. Cloud Services for sign-in in notifications. Using online data storage. Testing an App from usability perspective. Story boarding an app

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Deitel, P., & Deitel, H. (2017). Android how to program (3rd ed.). Upper Saddle River, NJ: Pearson Education. ISBN-13: 978-0-13-444430-7. Type: Textbook

Natural Language Processing

Credit Hours	3 (3-0)	**Prerequisites**	Artificial Intelligence

Course Introduction:

Natural Language Processing (NLP) is a rapidly developing field with broad applicability throughout the hard sciences, social sciences, and the humanities. The ability to harness, employ and analyze linguistic and textual data effectively is a highly desirable skill for academic work, in government, and throughout the private sector.

This course is intended as a theoretical and methodological introduction to a the most widely used and effective current techniques, strategies, and toolkits for natural language processing, with a primary focus on those available in the Python programming language

Course Objectives:

Upon completion of this course, students should be able to:

- Evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available.
- Employ literary-historical NLP-based analytic techniques like stylometry, topic modeling, syn setting and named entity recognition in their personal research.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Extract information from text automatically using concepts and methods from natural language processing (NLP) including stemming, n-grams, POS tagging, and parsing.	C2	Understanding
2. Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis).	C3	Create
3. Analyze the syntax, semantics, and pragmatics of a statement written in a natural language.	C4	Analyze
4. Apply machine-learning algorithms to natural language processing.	C3	Apply

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

Course Content:

What is Natural Language Processing? NLTK, Python 3 and the Jupyter Notebook Introduction to HPC. Textual Sources and Formats 1: "What's in a Text?". APIs, social media, Web Scraping. Tokenization, N-grams and Scriptio continua. Building your Corpus. Stemming and Lemmatization, Synsets and Hypernyms. POS Tagging and Stop words. Text "Features" and TF-IDF Classification. The "Words" in a "Text". Named Entity Recognition (NER). Sentiment Analysis. What Kind of Text is it? (Machine Learning Approaches to Textual Data). Strengths, Weaknesses, Correlations. What's in a Topic. Stylometry & Stylometric Analysis. Dendrograms, PCA scatterplots & k-means. Plotting the Text, Finding the Plot. Document Clustering and Word Vectors Doc2vec, Word2vec. Advanced Vector Analyses. Dependency Parsing. Constituency Parsing. The Worlds Beyond the Text.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)
2. Jacob Eisenstein. Natural Language Processing
3. Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning

Computer Vision

Credit Hours	3 (2-1)	**Prerequisites**	

Course Introduction:

This course aims to convey the nature of some of the fundamental problems in vision, and to explain a variety of techniques used to overcome them. Vision is a rapidly evolving area of computer science, and new and emerging approaches to these problems are discussed along with more "classical" techniques. Various vision problems are considered, including: feature detection in images, e.g. edge detection, and the accumulation of edge data to form lines; recovery of 3D shape from images, e.g. the use of a stereo image pair to derive 3D surface information; forming image mosaics; video surveillance techniques, e.g. tracking objects in video; motion detection in video images, e.g. counting number of moving objects in a video; recognizing and classifying objects in images, e.g. searching a video for a particular object. Several assignments will be given to enable the student to gain practical experience in tackling some of these problems.

Course Objectives:

Upon completion of this course, students should be able to:

- Recognize and describe both the theoretical and practical aspects of computing with images. Connect issues from Computer Vision to Human Vision.
- Describe the foundation of image formation and image analysis. Understand the basics of 2D and 3D Computer Vision.
- Become familiar with the major technical approaches involved in computer vision. Describe various methods used for registration, alignment, and matching in images.
- Get an exposure to advanced concepts leading to object categorization and segmentation in images.
- Build computer vision applications.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Analyze cognitive tasks including image classification, recognition, and detection	C3	Analyze
2. Conduct computer vision experiments and report results systematically	C4	Create
3. Explain the application of neural networks to computer vision	C2	Understanding
4. Make use of geometric camera models and multiple view geometry	C1	Knowledge

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

Course Content:

Introduction to computer vision. Camera Projection and Image Filtering. Thinking in Frequency. Sampling and Aliasing. Color. Interest points and corners. Local image features, SIFT. Model fitting, Hough Transform. RANSAC and transformations. Camera Calibration, Epipolar Geometry. Dense Stereo Correspondence. 3D Point Processing and Lidar. Transformer architectures. Semantic Segmentation. "Unsupervised" Learning and

Colorization. Big Data, Crowdsourcing. Classical recognition techniques and Deeper Deep Architectures. Optical Flow.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Computer Vision: Algorithms and Applications by Richard Szeliski. Available for free online. 2. Computer Vision: A Modern Approach (Second Edition) by David Forsyth and Jean Ponce. Available for free online. 3. Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Available for free online. 4. Multiple View Geometry in Computer Vision (Second Edition) by Richard Hartley and Andrew Zisserman.

Wireless Networks

Credit Hours	3 (3-0)	**Prerequisites**	Computer Networks

Course Introduction:

The course will introduce the basics of wireless communication, the evolution of modern wireless cellular networks and the design principles of cellular network. It will also demonstrate the architecture and protocols of GSM, GPRS and WCDMA cellular systems. The course will develop concepts of the emerging wireless networks: Wireless LAN, Wireless Mesh Networks, Wireless Personal Area Networks, Wireless Sensor Networks, WiMAX. The emphasis will also be on exploring research issues in emerging wireless networks.

Course Objectives:

Upon successful completion of this course, students will be able to:

- Understand the basics of wireless communication.
- Have adequate knowledge about the evolution of wireless systems.
- Understand the design of cellular systems in terms of frequency planning, system capacity and quality of communication.
- Have knowledge of supporting packet data traffic on cellular systems and enhancements for higher data rates.
- Able to carry research in emerging wireless networks.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. To develop the concept of systems thinking in the context of mobile and wireless systems	C3	Create
2. To develop knowledge of the interplay of concepts and multiple sub-disciplines in mobile and wireless systems	C3	Create
3. To gain knowledge and experience in applying various computation methods and algorithms as a part of software development	C1	Knowledge
4. To read and understand scientific research papers and present them in a seminar talk.	C2	Understanding

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

Course Content:

Introduction to Wireless Communication, Wired vs. Wireless Communication, Electromagnetic Signal, Time-Domain Concepts, Frequency-Domain Concepts, Channel Capacity, Signal-to-Noise Ratio, EM Spectrum, Design Challenges, Wireless Transmission, Signal Encoding and Modulation, Categories of Noise, Attenuation and other Impairments, Multiplexing, Transmission Mediums, Propagation Modes, Multipath Propagation, Types of Fading, Error Detecting and Correcting Techniques, Multiple Access Techniques, CSMA and Spread Spectrum, Evolution of Wireless Networks (1G Cellular Networks, 2G Cellular Networks, 2.5G Cellular Networks, 3G Cellular Networks, Limitation of 3G, 4G Objectives, Issues, QoS, Security, Multimedia Services and Applications, Convergence of Cellular and WLAN, Billing Issue, WLANS(IEEE802.11), WiMAX (IEEE802.16), Wireless PAN(IEEE802.15)), Fundamentals of Cellular Concepts(Cellular Concept, AMPS Architecture, Frequency Reuse, Locating co-channel cells, Channel Assignment Strategies, Handoff Strategies, Prioritizing Handoff, Practical Handoff Considerations, Co-channel Interference and Capacity, Adjacent Channel Interference and Capacity, Channel Planning for Wireless System, Trunking and Grade of Service, Measuring Traffic Intensity, Trunked Systems, Erlang Charts, Improving Coverage and Capacity, Cell Splitting, Sectoring, Repeaters for Range Extension, Microcell Zone Concept), Analog Mobile Phone System (Introduction, Architecture, System Overview, Call Handling, Air Interface, Supervisory Signals, N-AMPS), GSM: Global System for Mobile Communication (Introduction, System Architecture, Network Areas, Specifications, Subscriber Services, Mobility, Identifiers in GSM Network) Identifiers in GSM Network, Call

Routing in GSM, GPRS: General Packet Radio Service (Introduction, Architecture, Registration and Session Management, Routing Scenario in GPRS, Channels Classification, Protocol Architecture, Air Interface, Data Routing and Mobility, Uplink Data Transfer, Downlink Data Transfer, QoS in GPRS, EDGE Airlink), IS-136, IS-136 Channels and Specifications, CDMA One / IS-95 (Advantages and Drawbacks of CDMA Cellular, Mobile Wireless CDMA Design Considerations, IS-95 CDMA Forward Channel), Walsh Codes, IS-95 Reverse Link, EDGE: Enhanced Data Rate for GSM Evolution (Introduction, Modulation and Coding Schemes, Link Adaptation and Incremental Redundancy, Capacity Planning, Dynamic Abis Pool, Benefits), WCDMA / UMTS (Introduction, Service Classes in UMTS, UTRAN Architecture, Radio Interface protocol Architecture, Protocol Models for UTRAN, Logical Channels in WCDMA, Spreading and Scrambling, Transport and Physical Channels, Signaling, Physical Layer Procedures, Compressed Mode Measurements, Handover Measurements, WCDMA Packet Data Access, Transport Channels For Packet Data, Packet Scheduling Algorithms),
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Wireless Internet and Mobile Computing: Interoperability and Performance, Kwok & Lau, Wiley 2007, ISBN 97880847186796884 2. 20 Recipes for Programming PhoneGap: Cross-Platform Mobile Development for Android and iPhone, Jamie Munro, O'Reilly Media, 2012.

Multimedia Systems

Credit Hours	3 (3-0)	Prerequisites	

Course Introduction:

The course introduces students with the theory and principles of multimedia contents constituting the multimedia system. In this course, students will learn an introduction to Multimedia and Tools, Graphics and Image Data Representations, Color in Image and Video, Basics of Digital Audio, Lossless Compression Algorithms, Lossy Compression Algorithms, Image Compression Standards, Basic Video Compression Techniques, MPEG Video Coding MPEG-1, 2, 4-7; Basic Audio Compression Techniques, MPEG Audio Compression.

Course Objectives:

The main objectives of this course are:

- To understand the basic concepts, components, and tools of Multimedia Systems.
- To develop an understanding of the elements constituting the development of effective multimedia systems.
- To identify the evolution, latest trends, and state-of-the-art in multimedia technology, standards, and applications.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	Domain	BT Level*
1. Integrate the calculations of error and uncertainty as integral components of investigations.	C3	Create
2. Determine the optimal parameters for various multimedia encoding strategies given specific storage, bandwidth, and bit error rate criteria.	C2	Understanding
3. Learn efficient use of MATLAB/ OpenCV programming environments to design and develop software tools.	C1	Knowledge
4. Demonstrate main features of project work and answer critical questions during demo and oral sessions.	D1	Numeracy Skills

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

Course Content:

Introduction graphics/image representation. Sampling quantization digital audio basics. Color theory and video. Lossless compression methods. Lossy compression methods. The JPEG standard. Video compression. Audio compression. Content analysis: audio/visual content descriptors. Content analysis: introduction to multimedia search. Special topic - Cloud Computing. Sampling and Quantization. Entropy based coding. Intra- and Inter-frame coding. Video Compression / Content Analysis.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Fundamentals of Multimedia byZ. M. Li andM. S. Drew,Prentice Hall (2004), ISBN: 0-13-127256-X
2. Digital Multimedia by N. Chapman and J. Chapman. 2ndEdition, Wiley 2004, ISBN: 0-470-85890-7

3. The Technology of Video and Audio Streaming by David Austerberry, Focal Press; 2nd Edition (2004). ISBN-10: 0240805801
4. Multimedia Security: Watermarking, Steganography, and Forensics by Frank Y. Shih, CRC Press; 1st Edition (2012), ISBN-10: 1439873313
5. Multimedia Computing by Daniel Cunliffe and Geoff Elliott, Lexden Publishing Ltd (2005). ISBN-10: 1904995055
6. Multimedia Foundations: Core Concepts for Digital Design by Vic Costello, Ed Youngblood and Susan Youngblood, Focal Press; 1st Edition (2012). ISBN-10: 0240813944

Digital Image Processing

Credit Hours	3 (2-1)	**Prerequisites**	Computer Vision

Course Introduction:

This is a graduate-level introductory course on the fundamentals of digital image processing. The course will emphasize the general principles of image processing. It will extend the signals and systems knowledge of the students to two-dimensional signals. This is a very important course for any student who wants to do a senior project related to image processing. Towards the end of the course, some recent advances in this field will also be discussed to provide understanding about future directions.

Class lectures will be complemented with programming exercises in MATLAB

Course Objectives:

On completion of this course the students will be able to:

1. Demonstrate a knowledge of a broad range of fundamental image processing and image analysis techniques and concepts
2. Identify, Demonstrate, and apply their knowledge by analyzing image processing problems and recognizing and employing
3. Design and create practical solutions to a range of common image processing problems and to critically assess the results of their solutions, including shortcomings

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Develop a link between time and frequency domain analysis and image processing applications.	C2	Understanding
2. Develop ability to code various algorithms	C3	Create
3. combine existing image analysis techniques to solve emerging problems	C1	Knowledge

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

Course Content:

Applications of digital image processing. Elements of digital image processing system. Image perception. Sampling and quantization. Basic relationships between pixels. Point processing. Spatial filtering. Degradation models. Inverse filtering. Minimum mean square error (Wiener) filtering. Constrained least squares filtering. Image reconstruction from projections. Multiresolution expansion. Wavelet transforms in one dimension. Wavelet transforms in two dimensions. Elements of information theory. Lossless compression.

Lossy compression. Image compression standards. Detection of discontinuities. Segmentation by thresholding. Region based segmentation. Chain codes. Fourier descriptors. Moments.
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. A. Rosenfeld and A. C. Kak, < Digital Picture Processing >, Academic Press, 1982 2. W. K. Pratt, <Digital Image Processing>, 3rd Edition, John Wiley & Sons, Inc., (2001). 3. Y. Q. Shi and H. Sun, <Image and Video Compression>, CRC Press, 1st (1999) or 2nd (2008) edition

Fuzzy Logic			
Credit Hours	3 (3-0)	**Prerequisites**	
Course Introduction:			
Course Objectives:			
After successful completion of the course, the students are able to • Understand the basic ideas of fuzzy sets, operations and properties of fuzzy sets and about fuzzy relations. • Understand the basic features of membership functions, fuzzification process and defuzzification process. • Design fuzzy rule-based system. • Know about combining fuzzy set theory with probability to handle random and non-random uncertainty, and the decision-making process. • Gain the knowledge about fuzzy C-Means clustering.			
Course Learning Outcomes (CLOs):			
At the end of the course the students will be able to:		**Domain**	**BT Level***
1. To develop the fundamental concepts such as fuzzy sets, operations, and fuzzy relations.		C1	Knowledge
2. To lean about the fuzzification of scalar variables and the defuzzification of membership functions.		C2	Understanding
3. To learn three different inference methods to design fuzzy rule-based system.		C2	Understanding
4. To learn different fuzzy classification methods.		C3	Create
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain			
Course Content:			
Background, Uncertainty and Imprecision, Statistics and Random Processes, Uncertainty in Information, Fuzzy Sets and Membership, Chance versus Ambiguity. Classical Sets - Operations on Classical Sets, Properties of Classical (Crisp) Sets, Mapping of Classical Sets to Functions Fuzzy Sets - Fuzzy Set operations, Properties of			

Fuzzy Sets. Sets as Points in Hypercubes. Cartesian Product, Crisp Relations- Cardinality of Crisp Relations, Operations on Crisp Relations, Properties of Crisp Relations, Composition. Fuzzy Relations - Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Cartesian Product and Composition, Non-interactive Fuzzy Sets. Tolerance and Equivalence Relations - Crisp Equivalence Relation, Crisp Tolerance Relation, Fuzzy Tolerance and Equivalence Relations. Value Assignments - Cosine Amplitude, Max-min Method, Other Similarity methods. Classification by Equivalence Relations - Crisp Relations, Fuzzy Relations. Cluster Analysis, Cluster Validity, c-Means Clustering - Hard c-Means (HCM), Fuzzy c-Means (FCM). Classification Metric, Hardening the Fuzzy c-Partition, Similarity Relations from Clustering
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Ross T.J, Fuzzy Logic with Engineering Applications, 2nd Edition, John Wiley & Sons, 2004. 2. Yen J and Langari R, Fuzzy Logic Intelligence, Control, and Information, Pearson, 2009.

Expert Systems			
Credit Hours	3 (3-0)	**Prerequisites**	
Course Introduction:			
Project oriented course. Fundamentals of automated reasoning and deductive systems. Application of automated reasoning in mathematics, digital systems design (verification of hardware and software) and problem solving. Rule-based expert systems augmented with rule weighting, certainty factors, and fuzzy logic. Applications in technical systems synthesis, diagnostics, and process control. Probabilistic reasoning based on Bayesian belief networks. Applications of Bayesian networks in diagnostics and prediction. Project work involves hands-on experience with prevalent expert system shells (e.g., Prover9/Mace4, CLIPS, Fuzzy CLIPS, MATLAB, Hugin Lite).			
Course Objectives:			
On completion of course: • Students will be able to explain and describe the concepts central to the creation of knowledge bases and expert systems. • Students will be knowledgeable about the tools and the processes used for the creation of an expert system. • Student will know methods used to evaluate the performance of an expert system. • Students will be able to conduct an in-depth examination of an existing expert system with an emphasis on basic methods of creating a knowledge base. • Students will be able to examine properties of existing systems in a case-study manner, comparing differing approaches.			
Course Learning Outcomes (CLOs):			
At the end of the course the students will be able to:		**Domain**	**BT Level***

1. Define and describe expert system and its main constituents.	C1	Understanding
2. Distinguish class of problems suitable for solving with expert systems.	C4	Analyze
3. Breakdown the problem and select crucial parts.		
4. Assemble various parts of knowledge and skills in order to devise the approach to solution.	C2 C3	Create Create
5. Design and create expert system suitable for solving particular problem.	C3	Create

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

Course Content:

Scope of AI: games, theorem proving, natural language processing, vision & speech processing, robotics, expert systems, AI techniques-search knowledge, abstraction. Problem solving: State space search: Production Systems, Search space control: dept-first, breadth-first search, heuristic search – Hill climbing, best-first search, branch & bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis. Knowledge Representation: Predicate Logic: Unification, modus pones, resolution, dependency directed backtracking. Rule based Systems: Forward Reasoning: conflict resolution, backward reasoning: use of no backtracks. Structured. Knowledge Representation: Semantic Nets: slots, exceptions & default frames, conceptual dependency, scripts. Handling uncertainty on-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, fuzzy logic. Learning: Concept of learning, learning automation, genetic algorithm, learning by inductions, neural net. Expert Systems: Need & justification for Expert Systems, knowledge acquisition, Case Studies: MYCIN, RI.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations

Course Assessment:

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

Reference Materials:

1. Peter Jackson (1999.), Introduction to expert systems, 3rd Ed., Addison Wesley.
2. Wos, L., Overbeek, R., Lusk, E., Boyle (1992.), Automated reasoning: Introduction and Applications, McGraw-Hill
3. Finn B. Jensen (2010.), Bayesian Networks and Decision Graphs, 2nd. Ed., Springer Verlag.

Object Oriented Analysis and Design

Credit Hours	3 (3-0)	**Prerequisites**	Object Oriented Programming

Course Introduction:

Evolution of Object Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO program design process, classes, methods, objects, and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism, I/O and file processing, exception handling, UML (Conceptual model, class diagram system sequence diagram, etc.).

Course Objectives:

After the course, students should:

- Be able to use an object-oriented method for analysis and design
- Be able to analyze information systems in real-world settings and to conduct methods such as interviews and observations have a general understanding of a variety of approaches and perspectives of systems development, and to evaluate other IS development methods and techniques
- Know techniques aimed to achieve the objective and expected results of a systems development process
- Know different types of prototyping
- Know how to use UML for notation

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand principles of object-oriented paradigm.	C2	Understanding
2. Identify the objects & their relationships to build object-oriented solution	C3	Apply
3. Model a solution for a given problem using object-oriented principles	C3	Apply
4. Examine an object-oriented solution.	C4	Analyze

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

Course Content:

Principles of Object Technology, Introduction to UML, Unification, UML Diagrams, Unified Process & Rational Unified Process, RUP Disciplines, Case Study Analysis and Basics, Case Study, About Inception, Feasibility and Risk Analysis. Understanding Requirements, Requirement Types, Use Case Modeling: Use Case Writing Styles, EBP Guidelines. System Use Case Diagram, Use Case Table, Activity Diagram, Supplementary Specifications, Vision Document, Glossary, Rational Rose Overview, Use Case & Activity Diagram Modeling in Rational Rose. Elaboration Phase of RUP; Configuration Management; System Sequence Diagram, Domain Model: Identifying Business Classes, Associations. Implementation of System Sequence & Domain Model. Use Case Dependencies, Analysis Use Case Diagram, Implementation of Sequence, Collaboration, Analysis Use Case Diagram. State Chart Diagrams and Implementation. Design Patterns. Use Case Realization Using GRASP Patterns, Design Model: Determining Visibility. Modeling Generalization, Creating Design Class Diagram, Mapping Data Model to Domain Model. Implementation of Design Class Diagram, Coding patterns. Mapping Design to Code. More Patterns for Assigning Responsibilities, Polymorphism, Pure Fabrication, Indirection, Protected Variation. GoF Design Patterns: Adapter, Factory. Gof: Singleton, Strateg. Composition, and Façade and Discuss Remaining Patterns.

Teaching Methodology:

Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Booch G., "Object oriented analysis and design", Addison- Wesley Publishing Company 3rd edition. 2. Rambaugh J, Blaha.M. Premeriani, W., Eddy F and Loresen W., "ObjectOriented Modeling and Design", PHI 3. Martin Fowler, Kendall Scott, "UML Distilled", Addision Wesley 4. Eriksson, "UML Tool Kit", Addison Wesley.

Web Engineering			
Credit Hours	3 (3-0)	**Prerequisites**	Programming Fundamentals

Course Introduction:

Web Engineering is the application of systematic, disciplined, and quantifiable approaches to development, operation, and maintenance of Web-based applications

Course Objectives:

This course will address issues associated with large-scale web application development including requirements, architectural design and documentation, server and client-side development technologies, and service-oriented computing technologies. After completion of this course, students will be able

- To analyze, architect and design comprehensive systems for the creation, dissemination, storage, retrieval, and use of electronic records.
- To use some of the development languages, frameworks, and reusable services in order to manipulate information on the World Wide Web.
- To learn techniques and evaluation metrics for ensuring the proper operability, maintenance, and security of a web application.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Discuss how web standards impact software development.	C1	Knowledge
2. Describe the constraints that the web puts on developers.	C2	Understanding
3. Design and implement a simple web application.	C4	Analyze
4. Review an existing web application against a current web standard.	C4	Analyze

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

Course Content:

Web programming languages (e.g., HTML5, CSS 3, Java Script, PHP/JSP/ASP.Net), Design principles of Web based applications, Web platform constraints, Software as a Service (SaaS), Web standards, Responsive Web Design, Web Applications, Browser/Server Communication, Storage Tier, Cookies and Sessions, Input Validation, Full stack state management, Web App Security - Browser Isolation, Network Attacks, Session

Attacks, Large scale applications, Performance of Web Applications, Data Centers, Web Testing and Web Maintenance.
Teaching Methodology:
Lecturing, Written Assignments, Project, Report Writing
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Web Engineering, Rajiv Chopra, Prentice-Hall of India, 2016 2. Web Engineering, Emilia Mendes and Nile Mosley, Springer Verlag, 2010. 3. Web Engineering: A Practitioners' Approach, Roger S. Pressman, McGraw Hill, 2008. 4. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML, CSS, DOM, JavaScript 3rd Edition, O'Reilly Media 2007. 5. JavaScript: The Definitive Guide, 8th Edition, David Flanagan. O'Reilly Media. 2014.