Institute of Management Sciences, Peshawar

Course Outlines Catalogue

BS-Software Engineering

This document proposes to approve the course outlines for all the courses offered in BS-Software Engineering Program for the purpose of official publication of the same (Annexure-II). The Scheme of studies and the courses therein are already approved in the 14th meeting of the Academic Committee held on 12th of August 2022 and need no further deliberations (Annexure-I).

Table of Contents

1. SCHEME OF STUDIES, BS (SOFTWARE ENGINEERING)

Software Engineering is the application of a systematic, disciplined and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient and economical software by applying the principles and practices of engineering. The department aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evolution of software product.

1.1 Curriculum for BS (Software Engineering)

The curriculum for BS-Software Engineering was approved in the 14th Academic Committee meeting held on 12th of August 2022. Following are the areas along with total number of credit hours, which are required to cover to complete the degree:

1.1.1 Areas Covered in BS (SE)

Common Courses		
Course Group	**Min. No. of Credit Hours**	**Min. No. of Courses**
General Education	19	7
Institute Electives	12	4
Mathematics & Science Foundation	12	4
Computing Core	39	11
Common Courses	**82**	**26**
Domain Courses		
Software Engineering CORE (Compulsory) courses	24	8
Software Engineering SUPPORTING Courses	9	3
Software Engineering ELECTIVE courses	18	6
Domain Courses	**51**	**17**
Grand TOTAL	**133**	**43**

1.1.1.2 General Education Courses

Course Code	Course Title	Credit Hours	Contact Hours
CSC 301	Introduction to Info. & Comm. 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**

1.1.1.3 Institute Elective Courses

(Must be any four courses or 12 credit hours, not limited to the areas listed below, Institute 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
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
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
TOTAL		**12 (12-0)**	**12-0**

1.1.1.4 Mathematics and Science Foundation Courses

Course Code	Course Title	Credit Hours	Contact Hours
MTH 311	Calculus & Analytic 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 (12-0)**

1.1.1.5 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 57	Computer Networks	4 (3-1)	3-3
CSC 465	Operating System	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**

1.1.1.6 Software Engineering CORE (Compulsory) courses

Course Code	Course Title	Credit Hours	Contact Hours
CSC 461	Human Computer Interaction	3-0	3-0
SWE 523	Software Construction & Development	2-1	2-3
SWE 501	Software Design & Architecture	2-1	2-3
SWE 505	Software Project Management	3-0	3-0
SWE 425	Software Quality Engineering	3-0	3-0
SWE 621	Software Re-Engineering	3-0	3-0
SWE 421	Software Requirements Engineering	3-0	3-0
SWE 515	Web Engineering	3-0	3-0
TOTAL		**24 (22-2)**	**22-06**

1.1.1.7 Software Engineering SUPPORTING Courses

(THREE from the listed)

Course Code	Course Title	Credit Hours	Contact Hours
SWE 521	Business Process Engineering	3-0	3-0
SWE 801	Formal Methods in Software Engineering	3-0	3-0
STA 675	Operations Research	3-0	3-0
CSC 581	Modeling and Simulation	3-0	3-0
STA 651	Stochastic Processes	3-0	3-0
TOTAL		**12-0**	**12-0**

1.1.1.8 Software Engineering ELECTIVE courses

(Select any SIX courses from the following list)

(The list is by no means exhaustive. Institute may add new courses)

Course Code	Course Title	Credit Hours	Contact Hours
CSC 415	Functional Programming	3 (3-0)	3-0
CSC 421	Visual Programming	3 (3-0)	3-0
CSC 455	Theory of Automata	3 (3-0)	2-3
CSC 505	Real-Time Systems	3 (3-0)	2-3
CSC 511	Data Security and Encryption	3 (3-0)	2-3
CSC 515	Introduction to Bioinformatics	3 (3-0)	3-0
CSC 571	Mobile Application Development	3 (3-0)	2-3
CSC 601	Artificial Intelligence	3 (3-0)	3-0
CSC 611	Advanced Database Systems	3 (3-0)	3-0
CSC 615	Complex Networks	3 (3-0)	2-3
CSC 631	Cloud Computing	3 (3-0)	3-0
CSC 635	Distributed Computing	3 (3-0)	3-0
CSC 661	Data Mining	3 (3-0)	3-0
CSC 665	Data Warehousing	3 (3-0)	3-0
TOTAL (Any four courses or 12 credit hours)		**12 (X-X)**	**X-X**

1.2. Study Plan BS (SE)

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 and 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
ENG 302	English (Functional)	3 (3-0)	3-0	English (General)
HSS 305	Fundamentals of Islamic Studies	2 (2-0)	2-0	
CSC 321	Discrete Structures	3 (3-0)	3-0	
SWE 401	Software Engineering	3 (3-0)	3-0	
XXX XXX	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
STA 415	Probability & Statistics	3 (3-0)	3-0	
MTH 315	Linear Algebra	3 (3-0)	3-0	
SWE 421	Software Requirements Engineering	3 (3-0)	3-0	Software Engineering
XXXXXX	Institute Elective – II	3 (3-0)	3-0	
Total		**16(15-1)**	**15-3**	

Semester 4

Course Code	Course Title	Credit Hours	Contact Hours	Pre-requisite
CSC 451	Database Systems	4 (3-1)	3-3	
CSC 465	Operating Systems	4 (3-1)	3-3	
SWE 501	Software Design and Architecture	3 (2-1)	2-3	Software Requirements Engineering
CSC 461	Human Computer Interaction	3 (3-0)	3-0	Software Engineering
XXXXXX	Institute Elective – III	3 (3-0)	3-0	
Total		**17(14-3)**	**14-9**	

`

Semester 5

Course Code	Course Title	Credit Hours	Contact Hours	Pre-requisite
CSC 575	Computer Networks	4 (3-1)	3-3	
ENG 401	English (Academic)	3 (3-0)	3-0	
SWE 425	Software Quality Engineering	3 (3-0)	3-0	Software Eng., Software Requirements Engineering
XXX XX	SE Supporting – I	3 (3-0)	3-0	
XXX XX	SE Supporting – II	3 (3-0)	3-0	
Total		**16(15-1)**	**15-3**	

Semester 6

Course Code	Course Title	Credit Hours	Contact Hours	Pre-requisite
CSC 556	Information Security	3 (3-0)	3-0	
SWE 523	Software Construction & Development	3 (2-1)	2-3	Software Design & Architecture
SWE 515	Web Engineering	3 (3-0)	3-0	
XXX XXX	SE Elective ,I	3 (3-0)	3-0	
XXX XXX	SE Elective ,II	3 (3-0)	3-0	
XXX XX	SE Supporting - III	3 (3-0)	3-0	

Semester 7

Course Code	Course Title	Credit Hours	Contact Hours	Pre-requisite
SWE 505	Software Project Management	3 (3-0)	3-0	Software Engineering
SWE 621	Software Re-Engineering	3 (3-0)	3-0	Software Construction & Development
CSC 595	Professional Practices	3 (3-0)	3-0	
XXX XXX	SE Elective ,III	3 (3-0)	3-0	
XXX XX	SE Elective ,IV	3 (3-0)	3-0	
FYP 611	Final Year Project - I	3 (0-3)	0-9	
Total		**18(18-0)**	**18-0**	

Semester 8

Course Code	Course Title	Credit Hours	Contact Hours	Pre-requisite
XXX XXX	SE Elective ,V	3 (3-0)	3-0	
XXX XXX	SE Elective ,VI	3 (3-0)	3-0	
XXX XXX	Institute Elective ,IV	3 (3-0)	3-0	
FYP 612	Final Year Project - II	3 (0-3)	0-9	
Total		**12(9-3)**	**9-9**	

(Annexure-II)

2. Course outlines

Following are the detailed course outlines for all the courses mentioned in the above scheme of studies presented for the approval from the 15th Academic Committee meeting.

Advance Database Systems

Credit Hours	3 (3-0)	**Prerequisites**	Database Management 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 V's 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.

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:

1. 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.

2. Demonstrate teamwork skills/ ability to collaborate by working in groups on a laboratory experiment

3. 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.

4. 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, Toroids, Faraday’s experiments, Faraday’s Law of Induction, Lenz’s law, Motional emf, Induced electric field, Induced electric fields, The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves, Total internal reflection, Two source interference, Double Slit interference, related problems, Interference from thin films, Diffraction and the wave theory, related problems, Single-Slit Diffraction, related problems, Polarization of electromagnetic waves, Polarizing sheets, related problems.

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. Narciso Garcia, Arthur Damask, Steven Schwarz., “Physics for Computer Science Students”, Springer Verlag, 1998

Artificial Intelligence

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

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:

Artificial Intelligence (AI) is a constantly and actively growing and changing field. In this course, students will learn the basics of modern AI as well as some of the representative applications of AI.

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. Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 3rd edition, Prentice Hall, Inc., 2010.

2. Hart, P.E., Stork, D.G. and Duda, R.O., 2001. Pattern classification. John Willey & Sons.

3. Luger, G.F. and Stubblefield, W.A., 2009. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley.

Business Process Engineering

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

Course Introduction:

What is business process re-engineering? Business process re-engineering is a business management strategy focusing on analysis and design of workflows and processes within an organization. It aims to assist Organizations fundamentally re-think how they work to significantly improve customer service, reduce operational costs and become strong competitors.

Organizations undergo restructuring by focusing on the basic framework and design of their business processes. Broadly, there are three phases in the re-engineering process—planning, re-design and implementation. It has various elements of re-structuring too.

These include process-related, technology-related and Organization-related elements. Business process re-engineering is focused upon re-engineering around outcomes rather than tasks and helps link parallel activities rather than only connect the results.

Course Objectives:

This empowers students with:

- **The required skill, confidence and experience to partake in or independently drive business process re-engineering in your organization.**
- **The adequate skill set and exposure to partake in or drive business process re-engineering in any other organization, thus increasing the scope for career progression**
- **The capabilities and experience to drive implementation of an unconventional thought process among all employees of the organization**
- **The experience, confidence and knowledge to lay more focus on the customer, thus increasing client satisfaction and in turn market share and profitability for the organization.**

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. A proven best-practice method to achieve sustainable transformation.	C1	Knowledge
2. Apply a sophisticated toolkit of specific proven, high impact improvement tactics.	C2	Apply
	C4	Create

3. Develop compelling business-value driven business cases for change.		

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

Course Content:

Organize around outcomes, not tasks, Identify processes and prioritize as per re-design urgency, Integrate information processing into real work, Treat geographically dispersed resources as centralized, Link parallel activities instead of integrating only results, Put the decision point where work performed; build control into processes, Capture information once and at the source, Planning, Re-design, Implementation, Competitiveness comparison, Process quality management, Strategic capacity analysis, Critical success factors versus performance drivers analysis, Change management, Brown paper flowcharting, Process activity analysis, Noticeable pace and quality of response to customer needs, Structure focusing on customer, Increased market share and profitability, Improved cycle times, cost ratios and quality.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Business Process Reengineering: Text and Cases" by R.Radhakrishnan and S.Balasubramanian.
2. “Reengineering the Corporation: a Manifesto for Business Revolution” by Michael Hammer & James Champy.

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.	C1	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.

Complex Networks

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

Course Introduction:

The course provides an introduction to complex network theory and its applications in physics, biology, technology and social sciences. Basic graph theory and the statistical physics foundations as well as applications to real world networks will be covered. A hands-on approach to analytical and computational techniques for real world networks will be provided. Essential network models, e.g. small world networks, scale free networks, spatial and hierarchical networks will be discussed and methods to generate them with a computer will be covered. Different network visualization techniques and complex network tools will be explored as well. The course will cover three main branches of network science: 1.) Network structure, 2.) Dynamical processes on networks, and 3.) Network evolution.

Course Objectives:

On successful completion of this module the students will be able to:

- Define and calculate basic network graphic metrics.
- Describe structural features of socio-technical networks.
- Relate graphic properties to network functions and evolution.
- Relate local properties to global emerging patterns.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Navigate a consistent part of the present day literature on the structure and dynamics of complex networks	C1	Knowledge
2. Identify specific research problems in network science and get access to tools needed to solve them.	C3	Problem Solving

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

Course Content:

Community detection, Clubs and cores, Random walks, Synchronization, Epidemic Spreading, Social dynamics, Evolutionary games, Temporal networks, Multilayer networks, Higher-order interactions

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Complex Networks, an Algorithmic Perspective, By Kayhan Erciyes, Copyright Year 2015.
2. Networks: an introduction (Newman, Oxford)

Computer Networks

Credit Hours:	3+1	**Prerequisites:**	None

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	C	2
2. **Explain** the services and functions provided by each layer in the Internet protocol stack.	C	2
3. **Identify** various internetworking devices and protocols, and their functions in a network.	C	4
4. **Analyze** working and performance of key technologies, algorithms and protocols.	C	4
5. **Build** Computer Network on various Topologies	P	3

* 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:

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

Course Assessment:

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

Reference Materials:

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

4. Data Communication and Computer Networks, 5th Edition by Behrouz A. Forouzan

Data Mining

Credit Hours	3 (2-1)	**Prerequisites**	Advance Statistics, Introduction to Data Science

Course Introduction:

Data Mining has emerged at the confluence of artificial intelligence, statistics, and databases as a technique for automatically discovering hidden patterns in large datasets. The main purpose of this course is the ability to analyze and construct knowledge from data.

The aims of this course are to:

- Expand on the student's understanding and awareness of the concepts of data mining basics, techniques, and application.
- Introduce the concepts of Data Pre-processing and Summary Statistics.
- Introduce the concepts of Frequent Item Set Generation, Associations and Correlations measures.
- Introduce the concepts of Classification, Prediction, and Clustering algorithms.

Build on the programming and problem-solving skills developed in previous subjects studied by the student, to achieve an understanding of the development of Classification, Prediction, and Clustering applications.

Course Objectives:

The course introduces students with basic applications, concepts, and techniques of data mining and to develop their skills for using recent data mining software to solve practical problems in a variety of disciplines.

Course Learning Outcomes (CLOs)

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Apply preprocessing techniques on any given raw data.	C3	Apply
2. Select and apply proper data mining algorithm to discover interesting patterns	C3	Apply
3. Analyze and extract patterns to solve problems and point out how to deploy solution	C4	Analyze
4. Evaluate systematically supervised, semi supervised and unsupervised models and algorithms with respect to their accuracy	C4	Analyze

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

Course Content:
Introduction to data mining and basic concepts, Pre-Processing Techniques & Summary Statistics, Association Rule mining using Apriori Algorithm and Frequent Pattern Trees, Introduction to Classification Types, Supervised Classification (Decision trees, Naïve Bae Classification, K-Nearest Neighbors, Support Vector Machines etc.), Unsupervised Classification (K Means, K Median, Hieratical and Divisive Clustering, Kohonan Self Organizing maps), outlier & anomaly detection, Web and Social Network Mining, Data Mining Trends and Research Frontiers. Implementing concepts using Python
Teaching Methodology:
Lectures, Written Assignments, Projects Presentations
Course Assessment:
Sessional Exam, Home Assignments, Quizzes, Presentations, Final Exam
Reference Materials:
1. Jiawei Han & Micheline Kamber, Jian Pei (2011). Data Mining: Concepts and Techniques, 3rd Edition. 2. Sohrabi, M. K., & Azgomi, H. (2017). TSGV: a table-like structure-based greedy method for materialized view selection in data warehouses. Turkish Journal of Electrical Engineering & Computer Sciences, 25(4), 3175-3187. 3. Introduction to Data Mining. Charu C. Aggarwal (2015). Data Mining: The Textbook 4. D. Hand, H. Mannila, P. Smyth (2001). Principles of Data Mining. MIT Press.

Data Security and Encryption

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

Course Introduction:

This course introduces basic concepts in cryptography and computer security and discusses both their theoretical foundations and practical applications. Various threats, attacks and countermeasures including cryptosystems, cryptographic protocols and secure systems/networks will be addressed. The course will cover: brief history of cryptography, encryption (conventional and public key), digital signatures, hash functions, message authentication codes, randomness, unconditional and computational security, zero-knowledge protocols, secure e-commerce, group communication security, anonymity, key escrow. A few popular security mechanisms (e.g., Secure IP, SSL, PGP) will also be discussed.

Course Objectives:

The main objective of this course is:

1. Explaining the key security requirements aligning with type of threats and vulnerabilities that attack the security of information or database systems.
2. Presenting symmetric and asymmetric cryptographic systems and covering most important parts of cryptology through introducing many cryptography techniques and algorithms.
3. Describing the most important advance encryption theories aligning with the number theories that necessary as requirements.
4. Explaining the hash function as an application of cryptography aligning with the concept of message integrity and digital signature authentication.
5. Understand the issues involved in using asymmetric encryption to distribute symmetric keys.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Presenting the most important key security requirements that required for any security systems generally and specifically.	C1	Knowledge
2. Utilizing and code developing for encryption algorithms that required to achieve confidentiality key security.	C3	Understanding
3. Building an appropriate encrypting system that designed for specific key size and message length.	C4	Create
	C3	Analyze

4. Investigating the suitability of a hash function for verifying the message integrity and digital signature authentication.		

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

Course Content:

Symmetric Ciphers: Classical Encryption Techniques, Random bit generation and stream ciphering, Asymmetric Ciphering: Public Key Cryptography, Cryptography Data Integrity, Digital Signature: Elgamal Digital Signature Schame

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Cryptography and Network Security: Principles and Practice, Global Edition, 7/E, William Stallings, Pearson, ISBN-10: 1292158581.
2. Introduction to Cryptography: Principles and Applications. Hans Delfs & Helmut Knebl, Second Edition.

Data Structures and Algorithms

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

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:

A detailed study of Basic Structures commonly used in Data Processing, Implementation (in C++) and Applications of basic data structures, A Comparative study of different Sorting and Searching Techniques

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

Data Warehousing

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

Course Introduction:

This course focuses on Data Warehousing (DWH), which is a different way of storing RDBMS data in order to facilitate the efficient execution of multi-dimensional queries (called dimensional modeling). After a basic introduction to motivate the course, we will cover the DWH architectures and implementation models in detail. We will also cover Business Intelligence (BI) and discuss the nature of its tag team with DWH. Along with success stories, we will comprehensively implement all dimensional modeling activities in the lab, starting from raw DBMSs and ending at the dashboard level. One of the course objectives is to teach Python to the students and how to do DWH on big data.

Course Objectives:

- To provide students with in-depth knowledge, skills and understanding in the areas of Data Mining and Data Warehousing and a range of techniques, conceptual models and tools to develop into professionals in the areas of 'Data, Information and Knowledge Management', data mining approaches such as clustering, classification, regression etc. and its applicability in a wide range of application areas.
- To provide students with high-level operational skills in the use of state-of the art software for KD/DM and DW/DSS, based on understanding of basic principles and the use of real-world case studies.
- To provide students with independent exploratory and research skills, linked with abilities to synthesize, integrate and critically analyses and compare features of the Knowledge Discovery/Data mining/Business Intelligence/Data Warehousing area.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe the underlying concepts of data warehousing and mining. Understand the Data Mining Process and implement data mining process-based solutions.	C3	Understanding
2. Find valid patterns in test data using data mining experiments with test data.	C3	Problem Solving

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

Course Content:

This course will cover fundamentals of data mining and data warehousing. Topics will range from statistics to machine learning to database, with a focus on analysis of data sets exhibiting different distributions. The course content will be including the conceptual framework of data mining, descriptions and examples of standard methods used in data mining, and the role of data mining in real life application. Additionally, the course will provide limited exercises and practical experience with a data mining related research. Also covered are Data warehousing fundamentals, project planning, business requirements definition, dimensional modeling, technical architecture, physical configuration options, product selection physical database design, data staging process and techniques, end user applications, deployment, management and growth.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Data Warehousing Fundamental by Paulraj Pooniah
2. Data Mining: Practical Machine Learning Tools and Techniques by Witten, Frank and Hall

Database Systems

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

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

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

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 distributed 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 distributed computers.	C1	Knowledge
2. Write portable programs for distributed architectures using Message-Passing Interface (MPI) library	C2	Understanding
3. Analytical modelling and performance of parallel programs.	C3	Apply
4. Analyze complex problems with shared memory programming with open MP.	C4	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.
2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier.

English Academic

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

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	
2. Demonstrate ability to think critically.	C1	
3. Utilize information and digital literacy skills.	C3	

* 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.

English Functional

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

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:

The course aims to:

- 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	Individual and Team Work
2. Complete Academic Writing tasks using writing process and strategies according to genres	C5	Communication
3. Use Language Skills and Strategies in different situations, for a variety of functions	C5	Life-long Learning

* 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 every day 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
2. Reading. Upper Intermediate. Brain Tomlinson and Rod Ellis. Oxford Supplementary

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***
4. Enrich the thought and culture and provides us with the most important international vehicle of expression.	C1	1
5. Enhance English language skills of the students and develop their critical thinking.	C3	1
6. Demonstrate ability to think critically	C3	7

* 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. 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.
3. Smalley, R. L., M. K Ruetten and D. Kozyrev. 2001. Refining Composition Skills. 4th Ed. Heinle & Heinle Inc., Boston, MA, USA.

4. Vawdrey C. 1993. Practical Business English. 2nd Ed. Richard d Irwin Publishing, New York City, NY, USA.

Formal Methods in Software Engineering

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

Course Introduction:

Modern software development inevitably requires the design and analysis of a number of different artifacts. Formal methods allow the mathematically precise formulation of some of these artifacts. This course is an introduction to the use of formal methods for the specification, design, and automatic analysis of software-based systems. Z notational language would be used for system design and verification.

Course Objectives:

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

- Model various classes of distributed systems within appropriate formalisms
- Interpret and apply the formal languages of the formalisms for modeling distributed systems
- Apply specific techniques for the analysis and verification of distributed systems

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Define and state the Z notion of correct system execution.	C1	Knowledge
2. Distinguish between correct and incorrect system behavior.	C3	Understanding
3. Apply formal logic in expressing (desired) system behavior.	C3	Apply
4. Construct formal models of real systems suitable for verification.	C4	Create

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

Course Content:

Introduction to Formal methods, Introducing Z, Elements of Z, Logic and Using Predicates in Z, Schemas and Schema Calculus, Formal Reasoning, Case Studies in Z, Safety-Critical

Protection System, Modeling Large Systems, Computer Graphics and Computational Geometry, Rule-Based Programming, Graphical User Interface, Concurrency and Real-time, Refinement, Program Derivation, and Formal Verification, Converting Z into Code.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Paulson, Lawrence.*ML For The Working Programmer*. Paulson's book is an introduction to the ML programming language and functional programming in general. ML/SML figure predominantly in many formal systems, so a basic understanding of the language and the concepts of functional programming is necessary for understanding how formal tools actually work.
2. Miller & Srivas, Formal Verification of the AAMP5 Microprocessor. Miller & Srivas discuss the use of a formal verification system on a real project. Unlike the Wong paper, this one is more of a study of the role of a formal method in an engineering project, and takes a much higher-level view.
3. Kling, Robert. "Systems Safety, Normal Accidents And Social Vulnerability".
4. Kling's paper is actually an overview on system safety, but includes a subsection on formal verification and its role in safety.

Functional Programming

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

Course Introduction:

This is a first course in programming. It makes use of a programming language called Haskell, in which programs can be viewed as mathematical functions. This makes the language very powerful, so that we can easily construct programs that would be difficult or very large in other languages.

An important theme of the course is how to apply mathematical reasoning to programs, so as to prove that a program performs its task correctly, or to derive it by algebraic manipulation from a simpler but less efficient program for the same problem.

The course provides hands-on experience of programming through two lab exercises: the first one aims to make you acquainted with the mechanics of writing Haskell programs, and the second one tackles a more challenging programming task.

Course Objectives:

By the end of the course, the students:

- Master foundational techniques from the paradigm of functional programming.
- Be trained in using abstraction to structure programs.
- Be able to explain and use recursion in general, as well as know how to distinguish between recursive and iterative processes.
- Be able to write and use higher-order functions.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Write programs in a functional style;	C2	Create
2. Reason formally about functional programs;	C1	Knowledge
3. Use polymorphism and higher-order functions;	C3	Understanding
4. Reason informally about the time and space complexity of programs	C1	Knowledge

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

Course Content:

Programming in a functional language. Recursion. Abstract data types. Data directed programming, memoization, object oriented programming, lists and streams. The strengths and weaknesses of functional programming compared to imperativ programming. Semantics for evaluating function calls and interpreting functional programs.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Graham Hutton, Programming in Haskell (2nd edition), Cambridge University Press, 2016.
2. Richard Bird, Thinking Functionally With Haskell, Cambridge University Press, October 2014.

Fundamentals of Accounting

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

Course Introduction:

We all have applied the process of accounting on a daily basis at some personal level sub-consciously. If you have ever made personal budgets, managed your bills, have made projections about future income or have made financial plans for your future you have used the essence of accounting in your personal life.

It is a common misconception that accounting is only for accountants. Accounting is now recognized as a life skill that one uses in personal as well as commercial existence. Not being familiar with accounting principles and terminology can prove disastrous for any manager or businessperson.

It will be like driving a car blindfolded. You may be adept at driving but being unable to see the traffic or signals will ultimately lead you to crash. Accounting is considered as the spine of any business enterprise.

Course Objectives:

Upon successful completion of this course participants shall be able to achieve the following objectives:

- Realize the need for accounting information in different aspects of the business
- Understand the multi-facets of accounting and their application in personal as well as professional life
- Be adept at reading and interpreting financial statements
- Be aware of various accounting concepts and conventions
- Understand what financial statements include and how they are prepared
- Understand what and how accounting information is used in various management functions of planning, organizing, directing and controlling
- Understand how accounting skills help in tax planning
- Achieve coordination among various departments through the smooth flow of financial information

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Comprehend and review financial statements	C1	Knowledge
2. Understand technical jargons and terminology used in accounting	C3	Understanding
3. Keep track of expenses and revenues	C2	Problem Solving
4. Analyze financial statements to make sound investment decisions	C3	Analysis

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

Course Content:

Introduction to Financial Accounting, Users of Financial Statement, Double Entry System, Generally accepted accounting principles, Some Accounting Terminology, Accounting cycle, Chart of Accounts, The distinction between Capital Expenditure and Revenue Expenditure, Cash and accrual systems of accounting, Posting to General Ledgers, Journal Entries, Accounts Receivable / Accounts Payable Ledgers, Bank Reconciliation / Cash Reconciliation, Provisions, Depreciation, Inventory valuation, Balance Sheet, Income & Expenditure Account, Cash Flow Statement, Statement of Changes in Equity, Notes to Financial Statements, Ratio Analysis, Vertical P&L and Balance Sheet, Horizontal P&L and Balance Sheet.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Accounting Principles 10th edition Weygandt Kieso Kimmel

Fundamentals of Islamic Studies

Credit Hours	3 (3-0)	**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. 4. Aslan, Ednan, and Marcia K. Hermansen. Religious Diversity at School: Educating for New Pluralistic Contexts. Springer VS, Springer Fachmedien Wiesbaden GmbH, 2021.

Fundamentals of Pakistan Studies

Credit Hours	3 (3-0)	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 Team Work
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. 5. Mitra, Nayan, and Schmidpeter René. Corporate Social Responsibility in Rising Economies Fundamentals, Approaches and Case Studies. Springer International Publishing, 2020.

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:

1. 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 other domain.
2. 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.
3. 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. Andueza-López, B., & López-Plaza, M. (2020). The TV-production shift during the COVID-19 health crisis: How TV language changed as a state of alarm was enforced in Spain. *Tripodos*, 2(47), 161–172.
3. Arndt, S., Räty, V.-P., Nieuwenhuis, T., Keimel, C., Ibáñez, F.,Perkis, A. (2017). Enhancing use of social media in TV broadcasting. *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video* Hilversum, Netherlands (ACM), 51–56. https://doi.org/10.1145/3084289.3089923

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:

1. Build an understanding of the fundamental concepts of computer networking.

2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.

3. Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

4. 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

Introduction to Bioinformatics

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

Course Introduction:

This course is designed to give students both a theoretical background and a working knowledge of the techniques employed in bioinformatics. Emphasis will be placed on biological sequence (DNA, RNA, protein) analysis and its applications.

Course Objectives:

The student will be able to:

1. Become familiar with a variety of currently available genomic.
2. Compare and analyze biological sequences and how to interpret the results of their analyses. Assessment will be based upon performance on computer assignments and exam questions.
3. Learn how to construct phylogenetic trees based on biological sequence data. Assessment will be based upon performance on computer assignments and exam questions.
4. Perform elementary predictions of protein structure and function.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Account for and use methods in sequence bioinformatics such as sequence alignment, phylogenetic analysis and pattern recognition.	C1	Knowledge
2. Analyze and compile results of bioinformatics analyses critically.	C2	Analyze
3. Solve given biological problems by using appropriate bioinformatics methods and databases	C3	Problem Solving

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

Course Content:

Introduction, Review of DNA replication, transcription, and translation, Genome organization, Introduction to DNA and protein databases, data storage, file formats, information retrieval, Database queries, sequence retrieval, Creation of restriction endonuclease maps, Dot plots, Sequence alignment, Local alignment, Global alignment, Multiple alignments, Sequence alignments continued, Alignment scores, Statistical significance of database searches, Genetic distances, Distance based phylogenies, Phylogenetic tree construction, Phylogenetic tree

construction continued, Character based phylogenies, Consensus sequences, Finding genes and open reading frames in DNA sequences, Microarrays and the transcriptome, Microarray analysis and applications of microarrays, Introduction to proteomics, Prediction of protein structure and function, Prediction of protein structure and function continued, Comparative genomics, Comparative genomics continued, Future directions of bioinformatics
Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Practical Bioinformatics, 1st ed., Agostino, M., Garland Science, 2013 2. Lesk, Arthur M. Introduction to Bioinformatics. Oxford University Press, 2020.

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.	C1	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

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:
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

Modeling and Simulation

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

Course Introduction:

Modeling and simulation (M&S) is the use of models (e.g., physical, mathematical, or logical representation of a system, entity, phenomenon, or process) as a basis for simulations to develop data utilized for managerial or technical decision making.

Course Objectives:

The course will introduce the basic concepts of computation through modeling and simulation that are increasingly being used by architects, planners, and engineers to shorten design cycles, innovate new products, and evaluate designs and simulate the impacts of alternative approaches.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain the model classification at different levels. 2. Analyze complex engineering systems and associated issues (using systems thinking and modelling techniques) 3. Apply advanced theory-based understanding of engineering fundamentals and specialist bodies of knowledge in the selected discipline area to predict the effect of engineering activities. 4. Analyze the simulation results of a medium sized engineering problem	C1 C3 C4 C4	Knowledge Apply Analyze Analyze

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

Course Content:

Introduction to modelling and simulation, System analysis, Classification of systems, System theory basics, its relation to simulation, Model classification at conceptual, abstract, and simulation models levels, Methodology of model building, Simulation systems and languages, Means for model and experiment description, Principles of simulation system design, Parallel process modeling using Petri nets and finite automata in simulation, Models of queuing systems, Discrete simulation models, Model time, Simulation experiment control, Overview of numerical methods used for continuous simulation. System Dymola/ Modelica, Combined simulation, Special model classes, Models of heterogeneous systems, Cellular automata and simulation, Checking model validity, Verification of models, Analysis of simulation results, simulation results visualization, model optimization, generating, transformation, and testing of pseudorandom numbers with overview of commonly used simulation systems.

Teaching Methodology:

Lecturing, Written Assignments, Project, Final Exam

Course Assessment:

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

Reference Materials:

1. Modeling and Simulation, Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D., Springer-Verlag, 2014.
2. System design, modeling and simulation using Ptolemy II, Claudius Ptolemaeus, , Ver 2.0, Creative Commons Attribution-ShareAlike 3.0 Unported, 2014

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:

This is an introductory course on object oriented programming, which is designed to develop understanding of fundamental concepts of object-oriented programming. The course covers a number of basic and advanced object oriented concepts including classes, objects, inheritance, polymorphism, composition, encapsulation, templates etc. The course aims to illustrate the object oriented concepts and develop solutions using C++ and a little bit of JAVA language, their design principles and tools.

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++, 3rd Edition by Robert Lafore
4. Java: How to Program, 9th Edition by Paul Deitel
5. Beginning Java 2, 7th Edition by Ivor Horton
6. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu.

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:

This course has two components: a theory component to teach you the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation. In the theory component, you will learn about processes and processor management, concurrency and synchronization, memory management schemes, file system and secondary storage management, security and protection, etc. The practice component will complement the theory component through programming assignments illustrating the use and implementation of these 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

Course Assessment:

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

Reference Materials:

1. Operating Systems Concepts, 9th edition by Abraham Silberschatz
2. Modern Operating Systems, 4th edition by Andrew S. Tanenbaum
3. Operating Systems, Internals and Design Principles, 9th edition by William StallingsWu

Operations Research

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

Course Introduction:

Operations Research (also called Management Science) is the study of scientific approaches to decision-making. Through mathematical modeling, it seeks to design, improve and operate complex systems in the best possible way. The mathematical tools used for the solution of such models are either deterministic or stochastic. Students will learn very powerful modeling and solution techniques for decision-making problems that are used today by many successful companies to help them save/earn millions of dollars.

Course Objectives:

This course is designed:

- To introduce the students how to use variables for formulating complex mathematical models in management science, industrial engineering and transportation science.
- To provide the students with opportunity of using various software package for solving linear programming and integer programming models.
- To introduce the students to the use of basic methodology for the solution of linear programs and integer programs.
- To introduce the students to the basic concepts of polyhedral theory and valid inequalities and how to integrate the theory to the solution methods for integer programming.
- To introduce the students to the advanced methods for large-scale transportation and assignment problems.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Be able to understand the characteristics of different types of decision-making environments and the appropriate decision making approaches and tools to be used in each type.	C1	Knowledge
2. Be able to build and solve Transportation Models and Assignment Models.	C3	Application
3. Be able to design new simple models, like: CPM, MSPT to improve decision-making and develop critical thinking and objective analysis of decision problems.	C4	Analysis
4. Be able to implement practical cases, by using TORA, WinQSB.	A1	Practical

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

Course Content:

Introduction to operations research. Linear programming. Duality. Other algorithms for linear programming. The transportation and assignment problems. Dynamic programming. Integer programming.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. W. Winston, Operations Research, Duxbury Press. Operations Research: Applications and Algorithms, Wayne L Winston, Indian University, 7th edition, 2014
2. Price, Camille C., et al. Operations Research: A Practical Introduction. Chapman &amp; Hall/CRC, 2023.

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.	C2	Explanation
2. Analyze random variables, probability distributions and sampling distributions.	C4	Analyze
3. Apply different probability and statistics techniques in engineering problems	C3	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
4. Linton, Oliver B. Probability, Statistics and Econometrics. Academic Press, 2017.

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 C2	Knowledge Problem Solving
2. Distinguish between various fields of computing.	C3	Understanding
3. Describe the core of any profession.	C3	Analysis
4. Write and analyze software contracts as an employer or to an employer.	A2	Ethics
5. Know the business and professional environment of software house.		

* 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

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. The aim is to 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, 4th Edition, Tony Gaddis.
2. Starting out with Programming Logic & Degins, 4th 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, 2nd Edition by Guttag, John
6. Practice of Computing Using Python, 3rd Edition by William Punch & Richard Enbody
7. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel
   Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

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).

Software Construction & Development

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

Course Introduction:

Software construction is detailed creation of working, meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging. It is a software engineering discipline. Some of the specific tasks involved in construction are verifying that the groundwork has been laid so that construction can proceed successfully, determining how your code will be tested,designing and writing classes and routines, creating and naming variables and named constants, selecting control structures and organizing blocks of statements, unit testing, integration testing, and debugging your own code, reviewing other team members' low-level designs and code and having them review yours,polishing code by carefully formatting and commenting it,integrating software components that were created separately and tuning code to make it faster and use fewer resources.

Course Objectives:

This course is design to introduce fundamental principles and techniques of software development i.e.

- Easy to understand, error free and modifiable object oriented programs and small-scale systems.
- To deal with code complexity, changeability, reusability.
- To build individual programs as assignments and projects in teams to explicitly learn unit development and integration issues.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain important concepts of code by design; object oriented design translations into object oriented code i.e. abstraction, modularity, and concurrency, software design patterns, error free programming, unit and integration testing, debugging.	C1	Knowledge
2. Select and apply appropriate object oriented programming to constructs; optimized, error free, and reusable, code as individual developer.	C3	Application
3. Design and develop a small software system as a team by using appropriate programming, testing, and integration tools and techniques.	C3	Application

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

Course Content:

Introduction to Software Construction. Importance of Prerequisites of Target Software. Key Construction Decisions: Choice of Programming Language, Programming Conventions, Localization Aspects of Technology, Selection of Construction Practices. Design in Software Construction. Design Building Blocks. Defensive Programming. The Software-Quality Considerations. Collaborative Construction. Refactoring. Program Size & Software Construction. Managing Software Construction. Integration. Programming Tools. Layout and Style. Self-Documenting Code.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Hassan Gomaa, “Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures”, Cambridge University Press, 2017.
2. Craig Larmen, “Applying UML & Patterns: An Introduction to Object-Oriented Analysis & Design and Iterative Development” 3rd Edition. 3. Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates, “Head First Design Patterns”, O’Reilly Media, Inc., 2004.

Software Construction and Development

Credit Hours:	3 (2-1)	**Prerequisites:**	Software Design and Architecture

Course Learning Outcomes (CLOs):	
At the end of the course the students will be able to:	**Domain**
1. Create, Critique and Refine customer Use Cases.	C
2. Transform Use Cases into Object Oriented software Realizations through OO Analysis, OO Design and OO Coding (in Lab component).	C
3. Document your requirements, analysis, and design models in the Unified Modeling Language (UML) notation.	C
4. Apply techniques of state machines and design patterns to designs for implementation.	C

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

Course Content:

Introduction to Object Oriented Analysis and Design, Iterative development, Unified process ‘

Introduction to UML and Case Studies, Requirements Modeling using use cases,

use case diagrams and scenarios, Understanding the role of use cases in functional requirements and design, Interaction diagrams, sequence and collaboration diagrams, UML: Use case, Use case diagrams, Activity diagrams, Sequence Diagram (System level), Domain modeling, creating domain models from requirements, System Sequence Diagrams (SSDs), Operation contracts, System layers Design Class Diagrams, Method names, Multiobjects, Navigability, Dependency relationships, Reference attributes & Role names Gang of Four Patterns , Grasp Patterns and their application.

Teaching Methodology:

Lecturing, Written and Lab Assignments, Project, Report Writing

Course Assessment:

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

Reference Materials:

1. Object oriented Systems Analysis and Design Using UML (Second Edition) Simon Bennet, Steve McRobb 2002 ,McGraw-Hill
2. Object-oriented software engineering by Bruegge, Bernd | Allen H. Dutoit
   ISBN: 8129704331 Publication Date: 2004
3. Advanced Systems Design with Java, UML and MDA by Lano, Kevin Pages 386 Date Published May 2005

Software Design & Architecture

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

Course Introduction:

This course focuses on elementary concepts in software design especially the object-oriented software design. This course also focuses on the architectural design issues as well as the use of design patterns for solving different design problems. Use of modern object oriented design and analysis tools like UML will also be covered.

Course Objectives:

By the end of the course, students will learn:

- Software Architecture & Design Patterns are taught as reusable components of the design.
- Several architectural styles, middleware architectures are briefly studied with systems examples to help students understand the concept and offer quick practice
- After the completion of this course student should be able to;
- Describe all important concepts of Software Architecture and design
- Construct software architecture and OO design models (artifacts) for given complex problem in team.
- Investigate existing solutions i.e. architectural styles and software design patterns of a particular complex software system design problem for evaluation.
- Translate the architectural views into an implementable architectural model using CASE tools.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe all important concepts of Software Architecture and design	C1	Knowledge
2. Construct software architecture and OO design models (artifacts) for given complex problem in team	C5	Create
3. Investigate existing solutions i.e. architectural styles and software design patterns of a particular complex software system design problem for evaluation.	C3	Analyze
4. Translate the architectural views into an implementable architectural model using CASE tools.	C2	understanding

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

Course Content:

Software Life Cycle & importance of Software architecture, Software Architecture Styles & Patterns: (, Client Server, Pipe & Filters, Distributed computing architecture, N-Tier,

Component Based Architecture, Service oriented architecture), Overview of architecture viewpoints and quality attributes, Functional viewpoint and Information viewpoint, Use case view: (Use case Model, SSD, Domain Model, UI), Logical/Structural View: Role of Analyst & Designer, Logical/Structural View (contd..): Design Class Diagrams, Object Diagrams, Composite Structure Diagram, Process/behavior view: Activity Diagrams, Sequence Diagram, Communication diagram, Process/behavior view (contd..): State(machine) diagram, interaction overview, timing diagram, Software Design Patterns: Structural Patterns, Software Design Patterns: Creational Patterns, Software Design Patterns: Behavior Patterns, Implementation/ Developer view: Component Diagram, Package Diagrams, Deployment/Physical view: Deployment diagrams, network topology, Operational viewpoint, Introduction to Middleware architecture (CORBA, RMI, and OO middleware, message-oriented middleware)
Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
1. Complete Guide to Referencing and Avoiding Plagiarism (2) by Neville, Colin, Date Published: 2010 2. Ostrowski, Adrian, and Piotr Gaczkowski. Software Architecture with C++: Design Modern Systems Using Effective Architecture Concepts, Design Patterns and Techniques with C++20. Packt Publishing Ltd., 2021.

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:

1. List and describe the fundamental phases of the Software Development Lifecycle (SDLC)
2. Define and describe fundamental software engineering terminology and coding practices
3. Explore/explain relationships between software engineering and other engineering disciplines (Systems Engineering, Electrical and Computer Engineering, Industrial Engineering)
4. Modify/build a software program that introduces students to software development tools /environments
5. Troubleshoot and debug changes made to an existing software program
6. Develop an original Python software program, learning basic Python language syntax
7. 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., 10th Edition, Pearson Inc., 2014
2. Software Engineering, A Practitioner’s Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.
3. Software Engineering. South African Bureau of Standards, 2019.

Software Project Management

Credit Hours	3 (3-0)	Prerequisites	None

Course Introduction:

The course provides an in depth examination of project management principles and modern software project management practices. The five process groups and nine knowledge areas of the Project Management Institute Body of Knowledge (PMI BOK) are examined in the context of the systems development lifecycle. Methods for managing and optimizing the software development process are discussed along with techniques for performing each phase of the systems development lifecycle. Portfolio management and the use and application of software project management tools are also discussed.

Course Objectives:

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

- Understand the fundamental principles of software project management & have a good knowledge of responsibilities of project manager.
- Be familiar with the different methods and techniques used for project management.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Define the scope of software project management, the usual stages of software project, and the success criteria for a project.	C1	Knowledge
2. Describe the contents of a typical business plan and project portfolio management.	C2	Understanding
3. Plan for a project, Carry out an evaluation for a selection of projects using a variety of cost-benefit evaluation techniques and evaluate the business risks involved in a project.	C3	Create
4. Define qualities of good software and design methods of measuring the required qualities in software.	C1	knowledge

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

Course Content:

Planning, Tracking, & measurement ,Organizational structures ,Project charter ,Statement of Work ,Request for Proposal (RFP) ,Planning Phase ,Development lifecycle models ,Matching lifecycles to projects ,Project plans ,Project selection ,Project Financial Analysis ,Net Present Value, ,Return on Investment ,Payback Analysis Models ,Weight Scoring Model ,Estimation and Budgeting ,Effort Estimation ,Cost Estimation ,Scheduling ,Project network diagram fundamentals ,PERT techniques ,Gantt charts ,Critical chain scheduling ,Risk and Change

Management ,Risk management ,Change control ,More MS-Project ,Development Management ,Team models ,Requirements process ,Configuration management ,Software metrics ,Programming languages & tools ,Managing conflict and motivating ,Project Control ,Status reporting ,Project metrics ,Earned value analysis ,Communications Techniques ,Process Improvement ,MS Project: (a) Resource leveling (b) Other views ,System Test Process ,Test specifications ,Black box and white box testing ,Test scripts ,Unit and integration testing ,Acceptance test specifications ,Test tools ,Final Phases & Other Issues ,Project Recovery ,Documentation ,Cutover/Migration ,Post Project Reviews ,Closing ,Project Success ,Management support ,Expectations ,Success metrics

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. Jaspreet, Singh Harkawalpreet Singh Er. Software Project Management. Scholars' Press, 2021.
2. Leadership: Theory And Practice, 5th Edition, ISBN 978-1412974882,

Software Quality Engineering

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

Course Introduction:

This course introduces the student fundamental notions of software quality and the techniques used to build and check quality in software systems. A particular emphasis is placed on quantitative assessment of software quality and quality control using software-testing techniques. The students would not only be introduced with the theoretical background of these concepts but they would also be given hands-on experience of applying these concepts. The assignments would be planned carefully to enhance students' learning of applying the learnt concepts from practical standpoint.

Course Objectives:

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

- Introduce quality assurance and quality control techniques and develop a QA plan and Test Plan
- Document and report the findings
- Carry out inspections and carry out testing in a production environment.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Outline software testing and software quality assurance principles.	C1	Knowledge
2. prepare test case and test suites for completely testing all aspects of a system under test (SUT)	C3	Problem solving
3. compile findings of a quality assurance cycle	C5	Create

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

Course Content:

Basic Introduction. Software Quality Attributes. Introduction to Quality Engineering. Introduction to software testing. Software testing lifecycle. Testing Scopes. Testing Approaches. Testing Concepts. Introduction to testing process Requirement of software test planning. Testing documentation. Reporting and historical data recording. Testing philosophies. Testing strategies. Model based testing. Testing using models: Using finite state machine. Control-flow and dataflow based testing. Domain and combinatorial testing. Unit and integration testing. Integration testing. Slicing. Software reliability models and

engineering. Software inspections. Quality Models. Quality Measurements. System testing.
Teaching Methodology:
Lecturing, Written Assignments, Project, Practical Labs, Final Exam
Course Assessment:
Mid-Term Exam, Home Assignments, Quizzes, Presentation, Final Exam
Reference Materials:
• Galin, Daniel. Software Quality: Concepts and Practice. John Wiley &amp; Sons, 2018. • Paul Jorgensen, (2015), Software Testing, A Craftsman's Approach, Fourth Ed. CRC Press, Taylor and Francis Group • Bernard Homes, (2012). Fundamentals of Software Testing, ISTE, Wiley Publisher

Software Re-Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	Software Construction and Development

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain the concepts and technique of software re- engineering	C	1
2. Apply reengineering techniques to maintain and modify software systems	C	3
3. Analyze and understand maintenance related problems associated with object oriented software systems.	C	4
4. Able to perform complex design reengineering and reverse engineering problems.	C	5

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

Course Content:

Salient topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

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

Reference Materials:

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, 2016
2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc.2015
3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press 2000.

Stochastic Processes

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

Course Introduction:

A stochastic process is a set of random variables indexed by time or space. Stochastic modelling is an interesting and challenging area of probability and statistics that is widely used in the applied sciences.

Course Objectives:

Determine limit probabilities in Markov chains after an infinitely long period. Derive differential equations for time continuous Markov processes with a discrete state space. Solve differential equations for distributions and expectations in time continuous processes and determine corresponding limit distributions.

Course Learning Outcomes (CLOs)

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Define basic concepts from the theory of Markov chains and present proofs for the most important theorems.	C1	Knowledge
2. Compute probabilities of transition between states and return to the initial state after long time intervals in Markov chains.	C2	Understanding
3. Derive differential equations for time continuous Markov processes with a discrete state space.	C3	Apply
4. Solve differential equations for distributions and expectations in time continuous processes and determine corresponding limit distributions.	C4	Analyze

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

Course Content:

Discrete Markov chains, classification of states, first passage and recurrence times, absorption problems, stationary and limiting distributions. Chapman-Kolmogorov equations, Long run behavior of Markov chains, Absorption probabilities and expected times to absorption, Statistical aspects of Markov chains, The mover-stayer model, Application of a Markov chain and mover-stayer model to modeling repayment behavior of bank loans' grantees. Markov Processes in continuous time: Poisson processes, birthdeath processes. Poisson process The Kolmogorov differential equations, Limiting behavior of continuous time Markov chains The Q matrix, forward and backward differential equations, imbedded Markov Chain, stationary distribution. renewal theory, Brownian Motion and its generalizations, Discrete time martingales, Conditional expectation, Definition of a martingale and examples, Optional stopping theorem, Stochastic calculus

Teaching Methodology:

Lecturing, Written Assignments, Project, Final Exam

Course Assessment:

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

Reference Materials:

1. Jones, P.W. and Smith, P. (2017) Stochastic processes: An introduction, third edition. Boca Raton, FL: CRC Press.
2. Introduction to Probability Models, 11th Ed, Sheldon M. Ross, Academic Press 2014.
3. Essentials of stochastic processes, Durrett, Richard. Springer Science & Business Media, 2nd Ed, 2012.
4. Introduction to Stochastic Processes, 2nd Ed, G.F. Lawler, Chapman and Hall, Probability Series, 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. Introduction to computer theory, Daniel I. A. Cohen, 2nd Edition 3. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011 4. An Introduction to Formal Languages and Automata, by Peter Linz, 8th edition, Jones & Bartlett Publishers.

Visual Programming

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

Course Introduction:

This course introduces computer programming using the Visual Programming language with object-oriented programming principles. Emphasis is on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger. Upon completion, students should be able to design, code, test and debug at a beginning level. This course has been approved to satisfy the Comprehensive Articulation Agreement for transferability as a pre-major and/or elective course requirement.

Course Objectives:

This course will provide a managerial perspective of information systems and what role they play in an organization. Student learn about the modern technologies and how organizations can use these technologies for their growth.

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Use the different elements of a visual programming language as building blocks to develop correct, coherent programs.	C1	Knowledge
2. Program using the fundamental software development process, including design, coding, documentation, testing, and debugging.	C3	Application
3. Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs with VB.Net.	C4	Analysis

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

Course Content:

Different type of Visual Programming * Graphical User Interface * The need of Visual Programming * Rapid Application Development (RAD) Tools * Advantages of Visual Programming * Disadvantages of Visual Programming* Discuss the transformation in computing, internet and application development * Identify the need for .NET * Explain the role of CLR and Intermediate Language * Describe the core components of Microsoft .NET* Introduction to Class Libraries * Properties and Methods * Events and Event Handlers * Winforms GUI * Form (Properties, Methods and Events) * Controls in Winform * Dialog Boxes * Types of Dialog Boxes * Visual Effect in Winform * Exception * Types of Errors * Exception Classes * Properties of Exceptions * Handling Exceptions * ErrorProvider Control * Configuration Overview * Authentication and Authorization * Forms Authentication *

Windows Authentication * Security and ASP.NET.

Teaching Methodology:

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

Course Assessment:

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

Reference Materials:

1. J.C. Bradley, A.C. Millspaugh, "Programming in C# .NET", McGrawHill, 2014, ISBN 0-07-121564-6. Text Book.
2. Deitel and Deitel, "Visual C# : How to Program", 6/e Edition, Prentice Hall / Pearson Education, 2017, ISBN 978-0-13-650154-0.

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

1. To analyze, architect and design comprehensive systems for the creation, dissemination, storage, retrieval, and use of electronic records.
2. To use some of the development languages, frameworks and reusable services in order to manipulate information on the World Wide Web.
3. 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. Barrell, Dylan. Agile Accessibility Explained: A practical guide to sustainable accessible software development, Amazon Digital Services, 2019. 3. Ko, I.-Y., Murillo, J.M. and Vuorimaa, P. (2020) Current trends in web engineering: ICWE 2020 International Workshops KDWEB, sem4tra, and Wot4h Helsinki, Finland, June 9-12, 2020: Revised selected papers. Cham, Switzerland: Springer.