Bachelor of Science in Computer Science – BSCS

Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information [ref WordNet Princeton definition].

Computer Science 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 program aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evaluation of software product.

Computer Science spans a wide range, from its theoretical and algorithmic foundations to cutting- edge developments in robotics, computer vision, intelligent systems, bioinformatics, and other exciting areas. The overall scope of Computer Science may be viewed into the following three categories:

- To develop effective ways to solve computing problems. For example, Computer Science develops the best possible ways to store information in databases, send data over networks, and display complex images. The theoretical background offered by Computer Science allows determining the best performance possible, and their study of algorithms. It enables to develop new problem-solving approaches that provide better performance.
- It devises new ways to use computers intelligently and effectively. Progress in the areas of networking, database, and human-computer-interface came together as a result of the world-wide-web, which changed the entire world. Now, researchers are working to make robots that are practical aides and demonstrate intelligence, databases that create new knowledge and, in general, use computers to do new things.
- It deals with the design and implementation of software systems. Computer Science provides training and skills for the successful implementation of software systems that solve challenging programming jobs. Computer Science spans the range from theory to models, design and programming. Computer Science offers a comprehensive foundation that permits graduates to adapt to new technologies and new ideas.

Proposed Curriculum for
Bachelor of Science in Computer Science (BSCS) Program

Proposed curriculum (semester-wise course plan and contents of the courses) for BSCS Program is revised version of the existing program to align the program with the recommendations of NCEAC, HEC and Board of Studies of University of Karachi.

Summary Table for Total Number of courses offered and credit hours in the program - BSCS

Course Categories	Number of Courses	Credit Hours
Computing Core + Capstone Project	16	54
Domain Core	5	15
Domain Elective	5	15
Mathematics & Supporting Courses	4	12
General Education Requirement + Internship	15	39
Professional Certification	1	3
Total Number of Courses and Credit Hours	**47**	**138**

Semester-wise Course Plan

Semester – I		
Course Code	**Course Name**	**Credit Hrs.**
CS-351	Programming Fundamentals	4 (3+1)
CS-353	Introduction to Information & Communication Technologies	3 (3+0)
CS-355	Calculus and Analytical Geometry	3 (3+0)
CS-357	Applied Physics	3 (2+1)
CS-359	Functional English	3 (3+0)
CS-361	Islamic Studies or Ethics	2 (2+0)
CS-363	Pakistan Studies	2 (2+0)
Total Credits for Semester - I		**20**

Semester – II		
Course Code	**Course Name**	**Credit Hrs.**
CS-352	Object Oriented Concepts & Programming	4 (3+1)
CS-354	Digital Logic Design	4 (2+1)
CS-356	Linear Algebra	3 (3+0)
CS-358	Discrete Structures	3 (3+0)
CS-362	Ideology and Constitution of Pakistan	2 (2+0)
CS-364	Fehm e Quran	2 (2+0)
Total Credits for Semester – II		**18**

Semester – III		
Course Code	**Course Name**	**Credit Hrs.**
CS-451	Data Structures and Applications	4 (3+1)
CS-453	Software Engineering Fundamentals	3 (3+0)
CS-455	Computer Organization & Assembly Language	3 (2+1)
CS-457	Multivariable Calculus	3 (3+0)
CS-459	Probability & Statistics	3 (3+0)
CS-461	Urdu	2 (2+0)
Total Credits for Semester – III		**18**

Semester – IV		
Course Code	**Course Name**	**Credit Hrs.**
CS-452	Database Management Systems	4 (3+1)
CS-454	Expository Writing	3 (3+0)
CS-456	Theory of Automata	3 (3+0)
CS-458	Software Project Management	3 (3+0)
CS-460	Data Communication and Networking	3 (2+1)
CS-462	Professional Practices	3 (3+0)
Total Credits for Semester – IV		**19**

Semester – V		
Course Code	**Course Name**	**Credit Hrs.**
CS-551	Artificial Intelligence	3 (2+1)
CS-553	Operating Systems	4 (3+1)
CS-555	HCI and Computer Graphics	3 (2+1)
CS-557	Information Security	3 (2+1)
CS-559	Civics and Community Engagement	2 (2+0)
CS-561	Cloud Computing	3 (2+1)
Total Credits for Semester – V		**18**

Semester – VI		
Course Code	**Course Name**	**Credit Hrs.**
CS-552	Advance Database Management Systems	3 (2+1)
CS-554	Compiler Construction	3 (2+1)
CS-556	Design and Analysis of Algorithms	3 (3+0)
CS-558	Financial Accounting	3 (3+0)
CS-xxx	Domain Elective-1	3
CS-xxx	Domain Elective-2	3
Total Credits for Semester – VI		**18**

Semester – VII		
Course Code	**Course Name**	**Credit Hrs.**
CS-xxx	Domain Elective-3 **(Topic of Current Interest)**	3 (2+1)
CS-651	Parallel & Distributed Computing	3 (2+1)
CS-653	Entrepreneurship	3 (3+0)
CS-655	Final Year Project – I	3 (0+3)
CS-657	Internship	3 (3+0)
Total Credits for Semester – VII		**15**

Semester – VIII

Course Code	Course Name	Credit Hrs.
CS-xxx	Domain Elective-4	3 (2+1)
CS-xxx	Domain Elective-5	3 (2+1)
CS-654	Professional Certification	3 (0+3)
CS-656	Final Year Project – II	3 (0+3)
Total Credits for Semester – VII		**12**

Out of five domain elective courses two to be offered from following table in third year

Course Code	Domain Elective 1, 2	Credit Hrs.
CS-572	Advanced Programming	3 (2+1)
CS-574	Mobile Applications Development	3 (2+1)
CS-575	Web Technologies	3 (2+1)
CS-576	Multimedia Systems	3 (2+1)
CS-577	Simulation and Digital Twins	3 (2+1)
CS-578	Object Oriented Analysis and Design	3 (3+0)

Out of five domain elective courses three to be offered from following table in fourth/final year (seventh and eight semester). **Domain Elective 3, Topic of Current Interest** (Course will be offered from the list below or any contemporary topic in the domain).

Course Code	Domain Elective 3,4, 5	Credit Hrs.
CS-671	Software Testing & Quality Assurance	3 (3+0)
CS-672	Advanced Computer Graphics	3 (3+0)
CS-673	Cyber Security	3 (3+0)
CS-674	Data Science	3 (2+1)
CS-675	Data Warehousing and Data Mining	3 (2+1)
CS-677	Web Engineering	3 (2+1)
CS-678	Natural Language Processing	3 (3+0)
CS-679	Neural Networks and Fuzzy Logic	3 (3+0)

Course Outlines

Computing Core (54/138) 16 Courses (common to all computing programs)		
S#	**Course Title**	**Cr. Hrs.**
1.	Programming Fundamentals	4 (3+1)
2.	Object Oriented Programming	4 (3+1)
3.	Database Systems	4 (3+1)
4.	Digital Logic Design	4 (3+1)
5.	Data Structures	4 (3+1)
6.	Information Security	3 (3+0)
7.	Artificial Intelligence	3 (2+1)
8.	Data Communication and Networking	3 (2+1)
9.	Software Engineering	3 (3+0)
10.	Computer Organization & Architecture	3 (2+1)
11.	Operating Systems	4 (3+1)
12.	Analysis of Algorithms	3 (3+0)
13.	Theory of Automata	3 (3+0)
14.	Cloud Computing	3 (3+0)
15.	Final Year Project - I	3 (0+3)
16.	Final Year Project - II	3(0+3)

Course Name: Programming Fundamentals (CS-351)

Credit Hours: 4 (3-1)
Contact Hours: 3-3
Pre-requisites: None
Course Introduction:
This course provides fundamental concepts of programming to freshmen. The course is pre-requisite to many other courses; therefore, students are strongly advised to cover all contents and try to achieve CLOs to the maximum possible level. The course may be taught as language independent. Further, it is up to the university to choose any language for the practical/Lab purpose but that must be latest and market oriented.
Course Learning Outcomes:
1. Understand basic problem-solving steps and logic constructs
2. Apply basic programing concepts
3. Design and implement algorithms to solve real world problems
Course Outline:
Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, 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.
Reference Materials (or use any other standard and latest books):
1. Starting out with Programming Logic & Degins, Tony Gaddis
2. The C Programming Language, Brian W. Kernighan, Dennis M. Ritchie
3. Object Oriented Programming in C++ by Robert Lafore
4. C How to Program, Paul Deitel & Harvey Deitel
5. Problem Solving and Program Design in C++, Jeri R. Hanly & Elliot B. Koffman

---

Course Name: Object-Oriented Concepts & Programming (CS-352)
Credit Hours: 4 (3-1)
Contact Hours: 3-3
Pre-requisites: Programming Fundamentals
Course Introduction:
The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.
Course Learning Outcomes:
1. Understand principles of object-oriented paradigm.
2. Identify the objects & their relationships to build object-oriented solution
3. Model a solution for a given problem using object-oriented principles
4. Examine an object-oriented solution
Course Outline:
Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

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

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

Course Name: Database Management Systems (CS-452)
Credit Hours: 4 (3-1)
Contact Hours: 3-3
Pre-requisites: None

Course Introduction:

The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS concepts.

Course Learning Outcomes:

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

Course Outline:

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

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

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

---

Course Name: Digital Logic Design (CS-354)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: None

Course Introduction:

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

Course Learning Outcomes:

1. Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits
2. Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques
3. Apply the acquired knowledge to simulate and implement small-scale digital circuits
4. Understand the relationship between abstract logic characterizations and practical electrical implementations

Course Outline:

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

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

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

---

Course Name: Data Structures and Applications (CS-451)

Credit Hours: 4 (3-1)

Contact Hours: 3-3

Pre-requisites: Programming Fundamentals, OOC&P

Course Introduction:

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

Course Learning Outcomes:

1. Implement various data structures and their algorithms and apply them in implementing simple applications
2. Analyze simple algorithms and determine their complexities
3. Apply the knowledge of data structure to other application domains
4. Design new data structures and algorithms to solve problems

Course Outline:

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

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

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

---

Course Name: Information Security (CS-557)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: None

Course Introduction:

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

Course Learning Outcomes:

1. Explain key concepts of information security such as design principles, cryptography, risk management, and ethics
2. Discuss legal, ethical, and professional issues in information security
3. Apply various security and risk management tools for achieving information security and privacy
4. Identify appropriate techniques to tackle and solve problems in the discipline of information security

Course Outline:

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

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

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

Course Name: Artificial Intelligence (CS-551)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: Object Oriented Programming

Course Introduction:

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

Course Learning Outcomes:

1. Understand the fundamental constructs of Python programming language
2. Understand key concepts in the field of artificial intelligence
3. Implement artificial intelligence techniques and case studies

Course Outline:

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

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

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

Course Name: Data Communication and Networking (CS-460)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: None
Course Introduction:
This course introduces the basic concept of computer network to the students. Network layers, Network models (OSI, TCP/IP) and protocol standards are part of the course.
Course Learning Outcomes:
1. Describe the key terminologies and technologies of computer networks
2. Explain the services and functions provided by each layer in the Internet protocol stack
3. Identify various internetworking devices and protocols and their functions in a networking
4. Analyze working and performance of key technologies, algorithms and protocols
Course Outline:
Introduction and protocols architecture, basic concepts of networking, network topologies, layered architecture, physical layer functionality, data link layer functionality, multiple access techniques, circuit switching and packet switching, LAN technologies, wireless networks, MAC addressing, networking devices, network layer protocols, IPv4 and IPv6, IP addressing, sub netting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks.
Reference Materials: (or use any other standard and latest books):
1. Computer Networking: A Top-Down Approach Featuring the Internet by James F. Kurose and Keith W. Ross
2. Computer Networks by Andrew S. Tanenbaum
3. Data and Computer Communications by William Stallings
4. Data Communication and Computer Networks by Behrouz A. Forouzan

---

Course Name: Software Engineering Fundamentals (CS-453)
Credit Hours: 3 (3-0)
Contact Hours: 3-0
Pre-requisites: None
Course Introduction:
Software Engineering Fundamentals introduces students to the principles, processes, and practices of software development, covering the software development life cycle, design, implementation, testing, maintenance, and quality assurance.
Course Learning Outcomes:
1. Describe various software engineering processes and activates
2. Apply the system modeling techniques to model a medium size software system
3. Apply software quality assurance and testing principles to medium size software systems
4. Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis
Course Outline:
Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software process models, Agile software Development, Agile process models, Agile development techniques, Requirements engineering process, Functional and non-functional requirements, Context models, Interaction models, Structural models, behavioral models, model driven engineering, Architectural design, Design and implementation, UML diagrams, Design patterns, Software testing and quality assurance, Software evolution, Project management and project planning, configuration management, Software Process improvement
Reference Materials (or use any other standard and latest books):
1. Software Engineering, Sommerville, Pearson Inc.
2. Software Engineering, A Practitioner’s Approach, Pressman R. S.& Maxim B. R., McGraw-Hill

---

Course Name: Computer Organization and Architecture (CS-455)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: Digital Logic Design

Course Introduction:

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

Course Learning Outcomes

1. Acquire the basic knowledge of computer organization computer architecture and assembly language
2. Understand the concepts of basic computer organization, architecture, and assembly language techniques
3. Solve the problems related to computer organization and assembly language

Course Outline:

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

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

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

---

Course Name: Operating Systems (CS-553)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: Data Structures

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 Learning Outcomes:

1. Understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems
2. 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
3. Demonstrate the knowledge in applying system software and tools available in modern operating systems.

Course Outline:

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

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

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

---

Course Name: Design and Analysis of Algorithms (CS-556)

Credit Hours: 3 (3-0)

Contact Hours: 3-0

Pre-requisites: Data Structures

Course Introduction:

Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.

Course Learning Outcomes:

1. Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm
2. Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors.
3. Determine informally the time and space complexity of simple algorithms
4. List and contrast standard complexity classes
5. Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms
6. Use of the strategies (brute-force, greedy, divide-and- conquer, and dynamic programming) to solve an appropriate problem
7. Solve problems using graph algorithms, including single- source and all-pairs shortest paths, and at least one minimum spanning tree algorithm
8. Trace and/or implement a string-matching algorithm

Course Outline:

Introduction; role of algorithms in computing, Analysis on nature of input and size of input Asymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω, Sorting Algorithm analysis, loop invariants, Recursion and recurrence relations; Algorithm Design Techniques, Brute Force Approach, Divide-and-conquer approach; Merge, Quick Sort, Greedy approach; Dynamic programming; Elements of Dynamic Programming, Search trees; Heaps; Hashing; Graph algorithms, shortest paths, sparse graphs, String matching; Introduction to complexity classes.

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

1. Introduction to Algorithms by Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
2. Algorithm Design by Jon Kleinberg, Eva Tardos
3. Algorithms by Robert Sedgewick, Kevin Wayne

Course Name: **Theory of Automata (CS-456)**

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

Contact Hours: **3-0**

Pre-requisites: **None**

Course Introduction:

This course explores the mathematical foundations of computation through the study of formal languages, grammars, and automata. It introduces abstract machines such as finite automata, pushdown automata, and Turing machines to help students understand how problems are represented and solved computationally.

Course Learning Outcomes:

1. Explain and manipulate the different concepts in automata theory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc.
2. Prove properties of languages, grammars and automata with rigorously formal mathematical methods
3. Design of automata, RE and CFG
4. Transform between equivalent NFAs, DFAs and Res
5. Define Turing machines performing simple tasks
6. Differentiate and manipulate formal descriptions of languages, automata and grammars with focus on regular and context-free languages, finite automata and regular expressions.

Course Outline:

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.

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

1. Introduction to computer theory, Daniel I. A. Cohen. 2nd edition. (October 25, 1996)
2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich (September 28, 2007)
3. An Introduction to Formal Languages and Automata, by Peter Linz, Jones & Bartlett Publishers, 3rd edition. (October 1, 2000)
4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, New Age Publishers. Latest edition. (December 1, 2008)

Course Name: **Cloud Computing (CS-561)**

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

Contact Hours: **3-0**

Pre-requisites: **Operating System**

Course Introduction:

Cloud Computing introduces students to the concepts, architecture, and services of cloud-based systems. The course covers deployment models, service models (IaaS, PaaS, SaaS), virtualization, storage, networking, security, and resource management in cloud environments. Students learn how cloud computing enables scalable, flexible, and cost-effective solutions for modern applications and gain practical experience with popular cloud platforms and tools.

Course Learning Outcomes:

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

1. Describe fundamental cloud computing models and services (IaaS, PaaS, SaaS).
2. Deploy and manage applications in cloud environments.
3. Analyze the benefits and challenges of cloud computing.
4. Implement basic cloud security and compliance measures.
5. Evaluate cloud solutions for scalability, cost, and performance.

Course Outline:

Introduction to Cloud Computing, Cloud Computing Architecture, Cloud Service Models (IaaS, PaaS, SaaS), Deployment Models (Public, Private, Hybrid, Community), Virtualization and Virtual Machines, Cloud Storage and Databases, Cloud Networking, Cloud Security and Privacy, Resource Management and Scheduling, Load Balancing and Scalability, Cloud Application Development, Cloud Platforms and Tools (AWS, Azure, Google Cloud), Cloud Monitoring and Management, Case Studies and Real-World Applications.

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

1. Cloud Computing, Theory and Practice by Dan C. Marinescu, THIRD EDITION, Morgan Kaufmann Publishers, 2022
2. Cloud Computing Concepts, Technology & Architecture by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, Prentice Hall Publisher, 2013

BS - Computer Science Domain Core (18/138) 6 Courses		
Sr. No.	**Course Title**	**Cr. Hrs.**
1.	Advance Database Management Systems	3
2.	HCI & Computer Graphics	3
3.	Compiler Construction	3
4.	Parallel & Distributed Computing	3
5.	Software Project Management	3

Course Name: Advance Database Management Systems (CS-552)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: Database Management Systems

Course Introduction:

Advanced Database Management Systems is an extension to “Database Systems” course. The aim of the course is to enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficiencies of the centralized database systems. Moreover, it focuses to introduce the basic principles and implementation techniques of distributed database systems, and expose emerging research issues in database systems and application development.

Course Learning Outcomes:

1. Understanding advance data models, technologies and approaches for building distributed database systems.
2. Applying the models and approaches in order to become enabled to select and apply appropriate methods for a particular case
3. To develop a database solution for a given scenario/ challenging problem in the domain of distributed database systems.

Course Outline:

Introduction to advance data models such as object relational, object oriented. File organizations concepts, Transactional processing and Concurrency control techniques, Recovery techniques, Query processing and optimization, Database Programming (PL/SQL, T-SQL or similar technology), Integrity and security, Database Administration (Role management, managing database access, views), Physical database design and tuning, Distributed database systems, Emerging research trends in database systems, MONGO DB, NO SQL (or similar technologies)

Reference Materials:

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

---

Course Name: **HCI and Computer Graphics (CS-555)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites:

Course Introduction:

This course is an introduction to the fields of human-computer interaction (HCI) and computer graphics. It focuses on the principles and techniques of designing, implementing, and evaluating user interfaces for computer systems. The course also explores the principles and techniques of computer graphics, including 2D and 3D graphics, rendering, animation, and visual design. Students will learn how to design and implement user interfaces that are visually appealing, easy to use, and efficient

Course Learning Outcomes:

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

1. Explain key principles, models, and theories of human-computer interaction.
2. Analyze and identify usability issues in user interfaces through heuristic evaluation, cognitive walkthroughs, and user studies.
3. Design interactive user interfaces applying usability, accessibility, and visual design principles.
4. Describe fundamental graphics concepts such as rendering pipelines, transformation, clipping, and projection.
5. Implement 2D and 3D rendering algorithms including shading, lighting, rasterization, and texturing.

Course Outline:

It begins with an introduction to HCI concepts, human cognitive factors, perception, memory, and usability principles, followed by user-centered design processes, requirements gathering, prototyping techniques, and usability evaluation methods such as heuristic evaluation, cognitive walkthroughs, and user testing. The course then integrates computer graphics fundamentals, including coordinate systems, color models, geometric representations, and essential drawing algorithms for lines, circles, and polygons. Students explore 2D and 3D transformations, viewing pipelines, projection techniques, and clipping algorithms, along with illumination models, shading, texture mapping, and hidden-surface removal. The outline also includes interaction devices, interface styles, accessibility considerations, UI guidelines for web and mobile, and visual design principles that connect graphics with interaction design. Additional topics include animation basics, curves and surfaces, VR/AR interfaces, and the use of graphics libraries such as OpenGL for implementing interactive visual components. The course concludes with practical integration of HCI and graphics concepts through the design and development of interactive, user-centered graphical applications.

Reference Materials:

1. Interaction Design: Beyond Human–Computer Interaction by Helen Sharp, Yvonne Rogers, and Jenny Preece, 5th Edition, Wiley, 2019.
2. The Design of Everyday Things by Don Norman, Revised & Expanded Edition, Basic Books, 2013.
3. Computer Graphics with OpenGL by Donald Hearn, M. Pauline Baker, and Warren Carithers, 4th Edition, Pearson, 2014.
4. Computer Graphics: Principles and Practice by John F. Hughes et al., 3rd Edition, Addison-Wesley, 2014.
5. OpenGL Programming Guide (The Red Book) by Dave Shreiner et al., 9th Edition, Addison-Wesley, 2016.
6. Fundamentals of Computer Graphics by Steve Marschner and Peter Shirley, 4th Edition, CRC Press, 2016.

Course Name: **Compiler Construction (CS-554)**

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

Contact Hours: **2-3**

Pre-requisites: **Theory of Automata**

Course Introduction:

Compiler Construction is a core area of computer science that focuses on the design and implementation of software systems that translate high-level programming languages into executable machine code. This course provides students with both theoretical foundations and hands-on experience in building compilers, covering each major phase from lexical analysis to code generation and optimization. Students learn how programming languages are processed internally, how syntax and semantics are enforced, how intermediate representations are built, and how efficient target code is generated. By the end of the course, learners gain a deep understanding of how modern programming languages work beneath the surface, enabling them to design language features, build simple compilers, and better understand performance implications in real-world software systems.

Course Learning Outcomes:

1. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code Generation
2. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines
3. Design and implement a compiler using a software engineering approach
4. Use generators (e.g. Lex and Yacc)

Course Outline:

This course covers the design and implementation of compilers, beginning with an introduction to the role of compilers, the difference between compilation and interpretation, and the overall structure of a compiler. It then explores lexical analysis, including tokens, lexemes, regular expressions, finite automata, and the construction of lexical analyzers using tools like Lex or Flex. Syntax analysis is studied next, covering context-free grammars, parse trees, ambiguity, top-down parsing (LL(1)), bottom-up parsing (LR, SLR, LALR), and parser generation tools such as Yacc or Bison. The course continues with syntax-directed translation, semantic analysis, and intermediate code generation, including attribute grammars, type checking, symbol tables, scope rules, and three-address code. Topics on code optimization and target code generation are also included, addressing efficient machine code generation, basic block optimization, and peephole techniques. Finally, the course introduces runtime environments, memory management, and error detection and recovery, emphasizing practical compiler design principles, hands-on implementation experience, and real-world applications..

Reference Materials:

1. Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, 2nd Edition, Pearson, 2006.
2. Engineering a Compiler by Keith D. Cooper and Linda Torczon, 2nd Edition, Morgan Kaufmann, 2011.
3. Modern Compiler Implementation in C/Java/ML by Andrew W. Appel and Jens Palsberg, 2nd Edition, Cambridge University Press, 2002.
4. Programming Language Pragmatics by Michael L. Scott, 4th Edition, Morgan Kaufmann, 2015.

Course Name: Parallel and Distributed Computing (CS-651)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: Object Oriented Programming, Operating Systems

Course Introduction:
Parallel and Distributed Computing is an advance level programming course that helps to analyze sequential algorithms for possible modifications and implementation on available advance machines. This course will cover the study of various parallel and distributed computing hardware, operating system, algorithm design and implementation techniques in detail.

Course objectives:
Learn about parallel and distributed computers– Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library– Analytical modeling and performance of parallel programs– Analyze complex problems with shared memory programming with OpenMP

Course Learning Outcomes:
1. Learn about parallel and distributed computers.
2. Write portable programs for parallel or distributed-architectures using Message-Passing Interface (MPI) Library
3. Analyze complex problems with shared memory-programming with openMP

Course Outline:
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).

Reference Materials:
1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum and M. V. Steen, Prentice Hall. 4th edition. (February 26, 2016)
2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier. 1st edition. (October 31, 2011)

Course Name: Software Project Management (CS-458)
Credit Hours: 3 (3-0)
Contact Hours: 3 - 0
Pre-requisites:

Course Introduction:
Software Project Management focuses on planning, executing, and controlling software development projects to ensure they are completed on time, within budget, and according to quality standards. The course covers project lifecycle, estimation, scheduling, resource allocation, risk management, quality assurance, and team coordination, integrating both traditional and agile methodologies. Students gain practical insights into managing real-world software projects through case studies, tools, and best practices.

Course Learning Outcomes:
By the end of this course the student will be able to:
1. Explain fundamental project management concepts.
2. Apply project planning techniques to software projects.
3. Analyze risks and constraints in project execution.
4. Use project management tools to plan and track progress.
5. Demonstrate leadership and teamwork in simulated project environments.

Course Outline:

Introduction to Software Project Management, Project Management concepts, Project Management Tools, PMI‘s Knowledge areas, PMI Framework, PMI Process Groups. Understanding Organizations. Project Planning, Project Evaluation, Selection of an Appropriate Approach in Project, Software Effort Estimation, Activity Planning, Risk Management, Evaluating the Risks to the Schedule, Risk Control, Configuration Management and Maintenance, Environment for Configuration Control, Resource Allocation, Monitoring & Control, Review and Evaluation, Challenges of Outsourcing in Project Management

Reference Materials:

1. Software Project Management, Bob Hughes and Mike Cotterell, McGraw-Hill Education; 5th Edition (2017).
2. A Guide to the Project Management Body of Knowledge, Latest Edition,
3. Mastering Software Project Management: Best Practices, Tools and Techniques, Murali K. Chemuturi and Thomas M. Cagley Jr., J. Ross Publishing, 2010
4. Edition, published by Wiley, 2019.

BS - Computer Science Domain Elective (15/138) 5 Courses

Course Code	Domain Elective 1, 2	Credit Hrs.
CS-572	Advanced Programming	3 (2+1)
CS-574	Mobile Applications Development	3 (2+1)
CS-575	Web Technologies	3 (2+1)
CS-576	Multimedia Systems	3 (2+1)
CS-577	Simulation and Digital Twins	3 (2+1)
CS-578	Object Oriented Analysis and Design	3 (3+0)
Course Code	**Domain Elective 3,4, 5**	**Credit Hrs.**
CS-671	Software Testing & Quality Assurance	3 (3+0)
CS-672	Advanced Computer Graphics	3 (3+0)
CS-673	Cyber Security	3 (3+0)
CS-674	Data Science	3 (2+1)
CS-675	Data Warehousing and Data Mining	3 (2+1)
CS-677	Web Engineering	3 (2+1)
CS-678	Natural Language Processing	3 (3+0)
CS-679	Neural Networks and Fuzzy Logic	3 (3+0)

Course Name: Advanced Programming (CS-572)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: Object Oriented Programming, Programming Fundamentals, Data structures

Course Introduction:

Advanced Programming builds on foundational programming and object-oriented programming knowledge to equip students with the skills needed to develop complex, efficient, and maintainable software. The course emphasizes advanced topics such as dynamic data structures, recursion, multithreading, concurrency, database connectivity, design patterns, and event-driven programming, while providing hands-on experience through practical exercises, mini-projects, and case studies.

Course Learning Outcomes:

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

1. Apply advanced object-oriented and functional programming concepts to develop robust software solutions.
2. Develop multithreaded and concurrent programs with proper synchronization and performance considerations.
3. Utilize exception handling, file operations, and database connectivity in complex programs.
4. Apply design patterns, modular programming, and best practices to enhance code maintainability and reusability.
5. Analyze, debug, and optimize software for efficiency, scalability, and resource management.
6. Integrate advanced programming concepts into real-world mini-projects and case studies.

Course Outline:

Topics such as recursion, multithreading, concurrency, and synchronization techniques are covered to enable efficient and responsive programming. The course also introduces design patterns, modular programming, generic programming, advanced standard libraries, and event-driven programming. Emphasis is placed on writing clean, maintainable, and high-performance code, reinforced through practical exercises, mini-projects, and case studies that simulate real-world programming challenges.

Reference Materials:

1. *Data-Oriented Programming* by Yehonathan Sharvit, 1st Edition, O'Reilly Media, 2022.
2. *Algorithms and Data Structures for Massive Datasets* by Dzejla Medjedovic, Emin Tahirovic & Ines Dedovic, 1st Edition, Springer, 2022.
3. *Effective Java* by Joshua Bloch, 3rd Edition, Addison-Wesley, 2018.
4. *Clean Code: A Handbook of Agile Software Craftsmanship* by Robert C. Martin, 1st Edition, Prentice Hall, 2008.
5. *Head First Design Patterns* by Eric Freeman and Elisabeth Robson, 2nd Edition, O'Reilly Media, 2020.
6. *The Pragmatic Programmer: Your Journey to Mastery* by David Thomas and Andrew Hunt, 2nd Edition, Addison-Wesley, 2019.

Course Name: **Mobile Application Development (CS-574)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites: **Object Oriented Programming**

Course Introduction:

Mobile Application Development introduces students to the principles, tools, and techniques required to design, develop, and deploy applications for mobile platforms such as Android and iOS. The course covers mobile software architecture, user interface design, event-driven programming, data storage, and connectivity, along with best practices for performance optimization, security, and cross-platform development. Students gain hands-on experience building functional mobile apps while understanding the lifecycle, testing, and deployment considerations specific to mobile environments.

Course Learning Outcomes:

1. Discuss different architectures & framework for Mobile Application development.
2. Develop mobile applications using current software development environments.
3. Compare the different performance tradeoffs in mobile application development.

Course Outline:

Mobiles Application Development Platform; HTML5 for Mobiles; Android OS: Architecture, Framework and Application Development; iOS: Architecture, Framework; Application Development with Windows Mobile; Eclipse; Fragments; Calling Built-in Applications using Intents; Displaying Notifications; Components of a Screen; Adapting to Display Orientation; Managing Changes to Screen Orientation; Utilizing the Action Bar; Creating the User Interface; Listening for UI Notifications; Views; User Preferences; Persisting Data; Sharing Data; Sending SMS Messages; Getting Feedback; Sending E- mail; Displaying Maps; Consuming Web Services Using HTTP; Web Services: Accessing and Creating; Threading; Publishing, Android Applications; Deployment on App Stores; Mobile Programming Languages; Challenges with Mobility and Wireless Communication; Location-aware Applications; Performance/Power Tradeoffs; Mobile Platform Constraints; Emerging Technologies

Reference Materials:

7. Professional Android application development, Reto Meier, Wrox Programmer to Programmer.
8. iOS Programming: The Big Nerd Ranch Guide, Conway, J., Hillegass, A., & Keur, C.
9. Android Programming: The Big Nerd Ranch Guides, Phillips, B. & Hardy, B.

Course Name: **Web Technologies (CS-575)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites:

Course Introduction:

Web Technologies introduces students to the design, development, and deployment of modern web applications. The course covers fundamental web protocols, markup languages, client-side and server-side programming, database integration, and web application architecture. Students gain hands-on experience creating dynamic, interactive, and secure web applications while learning best practices in usability, performance, and maintainability.

Course Learning Outcomes:

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

1. Explain the principles and protocols underlying web technologies, including HTTP, HTML, CSS, and JavaScript.
2. Develop dynamic web applications using client-side and server-side programming techniques.
3. Apply best practices in web application architecture, scalability, and performance optimization.
4. Utilize modern frameworks, libraries, and tools to develop robust and maintainable web applications.
5. Understand web security principles and implement measures to protect web applications from common threats.
6. Evaluate and debug web applications to ensure functionality, usability, and cross-platform compatibility.

Course Outline:

Introduction to Web Applications, TCP/IP Application Services. Web Servers: Basic Operation, Virtual hosting, Chunked transfers, Caching support, Extensibility. SGML, HTML5, CSS3. XML Languages and Applications: Core XML, XHTML, XHTM MP. Web Service: SOAP, REST, WML, XSL. Web Services: Operations, Processing HTTP Requests, Processing HTTP Responses, Cookie Coordination, Privacy and P3P, Complex HTTP Interactions, Dynamic Content Delivery. Server Configuration. Server Security. Web Browsers Architecture and Processes. Active Browser Pages: JavaScript, DHTML, AJAX. JSON, Approaches to Web Application Development. Programing in any Scripting language. Search Technologies. Search Engine Optimization. XML Query Language, Semantic Web, Future Web Application Framework.

Reference Materials:

1. *Web Application Architecture: Principles, Protocols and Practices* by Leon Shklar and Richard Rosen, 2nd Edition, Wiley, 2009.
2. *Web Technologies: A Computer Science Perspective* by Jeffrey C. Jackson, 1st Edition, Prentice Hall, 2006.
3. *Designing with Web Standards* by Jeffrey Zeldman, 3rd Edition, New Riders, 2009.
4. *Web Programming and Internet Technologies: An E-Commerce Approach* by Paul J. Deitel and Harvey M. Deitel, 1st Edition, Pearson, 2014.
5. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5* by Robin Nixon, 5th Edition, O'Reilly Media, 2018.
6. *Modern Web Development: Understanding Domains, Technologies, and User Experience* by Dino Esposito, 2nd Edition, Microsoft Press, 2018.

Course Name: Web Engineering (CS-677)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: Programming Fundamentals

Course Introduction:

Web Engineering focuses on the systematic development, deployment, and maintenance of complex web applications. The course covers web application lifecycle models, requirements analysis, design methodologies, architectural patterns, and software quality attributes specific to web systems. Students learn about client-server architecture, web frameworks, content management, usability, security, performance optimization, and testing of web applications. Emphasis is placed on applying engineering principles and best practices to build scalable, maintainable, and reliable web-based systems through hands-on projects and case studies.

Course Learning Outcomes:

1. Discuss how web standards impact software development.
2. Describe the constraints that the web puts on developers.
3. Design and implement a simple web application.
4. Review an existing web application against a current web standard.

Course Outline:

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.

Reference Materials:

1. Web Engineering, Rajiv Chopra, Prentice-Hall of India, 2016
2. Web Engineering, Emilia Mendes and Nile Mosley, Springer Verlag, 2010.
3. Web Engineering: A Practitioners' Approach, Roger S. Pressman, McGraw Hill, 2008.
4. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML, CSS, DOM, JavaScript 3rd Edition, O'Reilly Media 2007.
5. JavaScript: The Definitive Guide, 8th Edition, David Flanagan. O'Reilly Media. 2014

Course Name: **Cyber Security (CS-673)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites:

Course Introduction:

This course introduces students to fundamental concepts and practices in cybersecurity, focusing on protecting information, systems, and networks from threats. Topics include basic security principles, information security terminology, malware types, and common attacks on server-side web applications such as SQL injection, cross-site scripting, and cross-site request forgery. Students also learn about planning and policy, network protocols, transport and network layer security, as well as securing wireless, cloud, and IoT environments. The course combines theoretical understanding with practical approaches to defend digital systems effectively.

Course Learning Outcomes:

1. Explain basic security concepts, terminology, and malware classifications.
2. Identify and mitigate common attacks on web applications, including SQL injection, XSS, and CSRF.
3. Understand network protocols, service models, and apply transport and network layer security measures.
4. Implement security measures for wireless networks, cloud services, and IoT devices.
5. Learn about basic security planning and policies to protect information and systems.

Course Outline:

Basic security concepts, Information security terminology, Malware classifications, Types of malwares. Server-side web applications attacks. Cross-site scripting, SQL Injection, Cross-site request forgery, Planning and policy, Network protocols and service models. Transport layer security, Network layer security, Wireless security, Cloud & IoT security.

Reference Materials:

1. CompTIA Security+ Guide to Network Security Fundamentals by Mark Ciampa —8th Edition, published by Cengage Learning, 2025.
2. Corporate Computer Security by Randall J. Boyle & Raymond R. Panko 5th Edition, published by Pearson Education, 2020.

Course Name: **Software Testing and Quality Assurance (CS-671)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

Software Testing and Quality Assurance introduces students to the principles, techniques, and practices for ensuring the quality and reliability of software systems. The course covers testing methodologies, quality assurance processes, verification and validation techniques, defect management, and the use of automated testing tools. Emphasis is placed on applying structured testing approaches, understanding software quality standards, and integrating testing and QA practices into the software development lifecycle to deliver robust, maintainable, and high-quality software.

Course Learning Outcomes:

1. Explain the principles of software testing and quality assurance, including standards and best practices.
2. Design and implement various testing strategies, such as unit, integration, system, and acceptance testing.
3. Apply automated testing tools and techniques to improve testing efficiency and coverage.
4. Identify, document, and manage software defects effectively.
5. Integrate quality assurance practices into the software development lifecycle to ensure reliable and maintainable software.

Course Outline:

Testing techniques. Black Box testing, White Box and Grey Box testing techniques. Quality Assurance planning and execution. Automated testing topics include constructing a framework, scripting techniques, generating a test data, generating test architecture, pre/post-processing, test maintenance, and job specific metrics. Current research topics in Software Testing and Quality Assurance.

Reference Materials:

1. Software Quality Assurance: Integrating Testing, Security, and Audit (Internal Audit and IT Audit), Abu Sayed Mahfuz, Auerbach Publications 2021.
2. Software Testing and Continuous Quality Improvement, published by O'Reilly Media, 2022.
3. Software Testing and Quality Assurance, authored by Deepak Jain & Ajay Kothari, published by BPB Publications, 2025.
4. Software Quality Assurance: From Theory to Implementation, authored by Daniel Galin, published by Pearson/John Wiley (latest reprint), 2018.
5. Introduction to Software Engineering, P Ammann and J Offutt, Cambridge University Press 2008.

Course Name: **Advanced Computer Graphics (CS-672)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

Advanced Computer Graphics focuses on modern techniques for creating high-quality visual content, covering advanced rendering, shaders, geometric modeling, real-time graphics, and physically based methods. Students build on foundational concepts to develop realistic effects and optimized graphical applications used in gaming, simulation, VR, and digital media.

Course Learning Outcomes:

1. Comprehend the structure of modern computer graphics systems
2. Explain the basic principles of implementing computer graphics fundamentals
3. Compare key algorithms for modelling and rendering graphical data
4. Develop design and problem-solving skills with applications to computer graphics
5. Construct interactive computer graphics programs using OpenGL

Course Outline:

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

Reference Materials:

1. Computer Graphics with Open GL (4th Edition) by Donald D. Hearn, Prentice Hall 2013
2. Foundations of 3D Computer Graphics by S. J. Gortler, The MIT press
3. Fundamentals of Computer Graphics by Steve Marschner , Peter Shirley, A K Peters /CRC Press (5th Edition), 2021.
4. Computer Graphics: Principles and Practice, by James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Addison Wesley, (3rd edition) 2013
5. Real-Time Rendering by Tomas Akenine-Möller, Eric Haines & Naty Hoffman — A K Peters / CRC Press, 4th Edition, 2018.

Course Name: **Object Oriented Analysis & Design (CS-578)**

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

Contact Hours: **3**

Pre-requisites: **Programming Fundamentals, OOC&P**

Course Introduction:

Object-Oriented Analysis and Design (OOAD) introduces students to the principles and practices of modeling, analyzing, and designing software using object-oriented methodologies. The course covers the full OO development lifecycle, including requirements engineering, use cases, prototyping, class and interaction diagrams, architectural and detailed design, state and activity diagrams, and object-oriented patterns. Students gain practical skills in applying OO concepts to develop scalable, maintainable, and well-structured software systems.

Course Learning Outcomes:

1. Apply object-oriented principles to model, analyze, and design software systems.
2. Perform requirements engineering using use cases and prototyping.
3. Develop and interpret class, interaction, and state diagrams for system behavior.
4. Apply design patterns and architectural principles to create maintainable software.
5. Verify and validate OO designs to ensure correctness, quality, and scalability.

Course Outline:

Principles of Object Technology. OOP Review. Principles of Modeling. OOA&D Overview. OO Development Process. Requirements Engineering, Analysis, and Specification: Requirements Engineering, Use Cases, Prototyping, Class Models. Interaction Diagrams. Verification and Validation. Architectural and Detailed Design. Class Diagrams. Interaction Diagrams. State Machines and Diagrams. Implementation, Package Diagrams. Activity Diagrams. OO Patterns, Verification and Validation. Note: Students may also be introduced to Object Diagram, Component Diagram, Package Diagram, Deployment Diagram, Network Diagram.

Reference Materials:

1. Applying UML and patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development by Craig Larman, Prentice Hall; 3rd Edition (October 30, 2004). ISBN-10: 0131489062
2. Using UML: Software Engineering with Objects and Components by Perdita Stevens, Addison-Wesley; 2nd Edition (February 13, 2006). ISBN-10: 0321269675
3. Fundamental of Object-Oriented Design in UML by Meiler Page-Jones, Addison Wesley, 2000. ISBN: 020169946X.
4. The Unified Modeling Language User Guide by G. Booch, J. Rambaugh and I. Jakobson, Addison-Wesley Professional; 2nd Edition (2005). ISBN- 10: 0321267974.

Course Name: **Simulation and Digital Twins (CS-577)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites:

Course Introduction:

This course introduces students to the principles and applications of simulation and digital twin technology in computer science. Students learn how to model, implement, and analyze digital representations of real-world systems, integrating interactive elements, physics, and real-time behaviors. Emphasis is placed on practical skills using modern simulation frameworks to develop, evaluate, and refine digital twin models for various applications including engineering, healthcare, and IoT systems.

Course Learning Outcomes:

1. Explain the principles of simulation and digital twin technology and their applications in computer science.
2. Design and implement digital twin models or simulation scenarios using simulation frameworks.
3. Integrate interactive elements, physics, and real-time behavior into simulations or digital twins.
4. Analyze simulation and digital twin performance, accuracy, and user interaction for refinement.

Course Outline:

overview of simulation principles and digital twin concepts, including types, architectures, and application areas. Students study modeling techniques and design methodologies for creating accurate and scalable digital twins. The curriculum covers simulation frameworks, interactive element integration, physics-based modeling, and real-time behavior implementation. Students learn methods for validating, verifying, and analyzing simulation and digital twin performance, accuracy, and user interaction.

Reference Materials:

1. *Digital Twins for Simulation-Based Decision-Making* — edited by Vinay Kulkarni, Tony Clark & Balbir S. Barn, 1st Edition, Springer Cham, 2025
2. *Simulation Techniques of Digital Twin in Real-Time Applications: Design, Modeling and Implementation* — edited by Abhineet Anand, Anita Sardana & Abhishek Kumar, 1st Edition, Wiley-Scrivener, 2024
3. *The Engineering of Digital Twins* — edited by John Fitzgerald, Cláudio Gomes & Peter Gorm Larsen, 1st Edition, Springer Cham, 2024
4. *Digital Twin: Fundamentals and Applications* — edited collection, 1st Edition, Springer, 2024
5. *Digital Twin – Fundamental Concepts to Applications in Advanced Manufacturing* — Surjya Kanta Pal, Debasish Mishra, Arpan Pal, Samik Dutta, Debashish Chakravarty, Srikanta Pal, 1st Edition, Springer Cham, 2022
6. *The Big Book of Simulation Modeling — Multimethod Modeling with AnyLogic 8* — Andrei Borshchev & Ilya Grigoryev, 1st Edition, AnyLogic, (latest printing) — practical guide for simulation modeling (agent-based, system dynamics, discrete-event)

Course Name: **Multimedia Systems(CS-576)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites:

Course Introduction:

Multimedia Systems introduces students to the principles, design, and implementation of systems that integrate text, graphics, audio, video, and animation. The course covers multimedia data types, compression techniques, storage and retrieval, multimedia communication, and system architectures. Students learn about authoring tools, interactive multimedia applications, and best practices for developing efficient, high-quality multimedia content. Hands-on exercises and projects enable students to design, implement, and evaluate multimedia systems for educational, entertainment, and business applications.

Course Learning Outcomes:

5. Understand and describe different multimedia data types, formats, and standards.
6. Apply multimedia compression, storage, and retrieval techniques effectively.
7. Design and develop interactive multimedia applications using appropriate tools and technologies.
8. Analyze multimedia system architectures and implement efficient multimedia communication.
9. Evaluate multimedia systems for usability, performance, and quality across various application domains.

Course Outline:

Introduction to multimedia systems, covering the fundamentals of multimedia data types including text, graphics, audio, video, and animation. Students learn about multimedia storage, retrieval, and compression techniques, emphasizing algorithms for efficient representation and transmission. The course explores multimedia communication, system architectures, and networking considerations for real-time and distributed multimedia applications. Authoring tools, software frameworks, and development environments are introduced to create interactive multimedia content. Topics also include multimedia synchronization, quality assessment, usability, and evaluation of multimedia systems.

Reference Materials:

1. Fred Halsall, "Multimedia Communications: Applications, Networks, Protocols, and Standards", Latest Ed.
2. Puri, "Multimedia Systems, Standards and Networks", Marcel Dekker, Latest Ed.
3. Steve Heath, "Multimedia and Communication Technology", Focal Press, Latest Ed.
4. Bill Whyte, "Multimedia Telecommunication", Chapman and Hall, Latest Ed.

Course Name: **Data Science (CS-674)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

Introduction to Data Science provides students with a foundational understanding of data science principles, techniques, and applications. The course covers the complete data science workflow, including data collection, cleaning, exploration, and analysis. Students gain hands-on experience performing exploratory data analysis (EDA) and applying basic machine learning algorithms to solve real-world problems. Emphasis is placed on understanding the data science process, interpreting results, and making data-driven decisions.

Course Learning Outcomes:

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

1. Understand the fundamental principles of data science.
2. Apply EDA and the Data Science processes.
3. Apply basic machine learning algorithms to solve real-world problems of moderate complexity.

Course Outline:

Fundamentals of Data Science, Data Collection and Sources, Data Preprocessing and Cleaning, Exploratory Data Analysis (EDA), Data Visualization Techniques, Basic Machine Learning Algorithms (Regression, Classification, Clustering), Model Evaluation and Validation, Data Science Process and Workflow, Real-world Applications and Case Studies, Tools and Frameworks for Data Science

Reference Materials:

1. Python for Data Analysis, by Wes McKinney, 3rd Edition, O'Reilly Media, 2022.
2. An Introduction to Statistical Learning with Applications in R, by Gareth James, Daniela Witten, Trevor Hastie & Robert Tibshirani, 2nd Edition, Springer, 2021.
3. Computational and Inferential Thinking: The Foundations of Data Science, by Ani Adhikari & John DeNero, 1st Edition, 2019.
4. Data Mining and Analysis: Fundamental Concepts and Algorithms, by Mohammed Zaki & Wagner Meira Jr., 1st Edition, Cambridge University Press, 2014.
5. Data Science from Scratch, by Joel Grus, 2nd Edition, O'Reilly Media, 2019.
6. Doing Data Science, by Cathy O'Neil & Rachel Schutt, 1st Edition, O'Reilly Media, 2013.
7. Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications, by Laura Igual, 1st Edition, Springer, 2017.

Course Name **Data Warehousing and Data Mining (CS-675)**
Credit Hours **3 (2+1)**
Contact Hours **3**
Pre-requisites

Course Introduction:

This course gives an introduction to methods and theory for development of data warehouses and data analysis using data mining. Data quality and methods and techniques for preprocessing of data. Modeling and design of data warehouses. Algorithms for classification, clustering and association rule analysis. Practical use of software for data analysis

Course Learning Outcomes:

1. Data preprocessing and data quality.
2. Modeling and design of data warehouses.
3. Algorithms for data mining Skills:
4. Be able to design data warehouses.
5. Ability to apply acquired knowledge for understanding data and select suitable methods for data analysis.

Course Outline:

Data Warehouse: (a) Data Model for Data Warehouses. (b) Implementing Data Warehouses: data extraction, cleansing, transformation and loading, data cube computation, materialized view selection, OLAP query processing. Data Mining: (a) Fundamentals: data mining process and system architecture, relationship with data warehouse and OLAP systems, data pre-processing. (b) Core topics include association rule mining, classification, clustering, regression, and anomaly detection. The course also explores data evaluation and visualization techniques for interpreting mined patterns. (c) Advanced topics include big data analytics, privacy-preserving data mining, and the use of modern data mining tools and frameworks for decision support and business intelligence.

Reference Materials:

1. Data Mining and Data Warehousing: Principles and Practical Techniques, june 2019, Cambridge University Press (June 27, 2019)
2. Data Warehousing, Data Mining, and OLAP (Data Warehousing/Data Management), McGraw-Hill (1997)

Course Name **Natural Language Processing (CS-678)**
Credit Hours **3 (2+1)**
Contact Hours **3**
Pre-requisites

Course Introduction:

This course introduces students to Natural Language Processing (NLP), covering fundamental concepts, text preprocessing, language modeling, semantics, and machine learning approaches. Students learn to develop practical NLP applications such as text classification, information extraction, machine translation, and dialogue systems using modern tools and neural network models.

Course Learning Outcomes:

1. Explain fundamental concepts of natural language processing, computational linguistics, and text preprocessing techniques such as tokenization, stemming, lemmatization, and POS tagging.
2. Apply language modeling, deterministic and stochastic grammars, parsing algorithms, and semantic representations to process and analyze text.
3. Implement corpus-based methods, n-grams, HMMs, smoothing techniques, and POS tagging for practical NLP tasks.
4. Develop solutions for information retrieval, text classification, information extraction, machine translation, and dialogue systems using statistical and machine learning approaches.
5. Utilize modern neural network architectures, word embeddings, and large language models (LLMs) with Python libraries (NLTK, spaCy, Hugging Face Transformers) to build NLP applications.

Course Outline:

Introduction to NLP, Computational Linguistics, Text Preprocessing (Tokenization, Stemming, Lemmatization, POS Tagging), Language Modeling, Deterministic and stochastic grammars, Parsing algorithms, CFGs, Representing meaning / Semantics, Semantic roles, Temporal representations, Corpus-based methods, N-grams and HMMs, Smoothing and backoff, POS tagging and morphology, Information retrieval, Vector space model, Precision and recall, Information extraction, Language translation, Text classification, categorization, Bag of words model, Statistical and Machine Learning Approaches in NLP, Sequence Labeling, Word Embeddings, Neural Networks for NLP (RNNs, LSTMs, Transformers), Question Answering, Machine Translation, Dialogue Systems, Large Language Models (LLMs), Practical Implementation with Python (NLTK, spaCy, Hugging Face Transformers)

Reference Materials:

1. Python Machine Learning, Sebastian Raschka. Publisher: Packt Publishing, (3rd Edition) 2019.
2. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit Latest Edition, Steven Bird, Ewan Klein and Edward Loper Publisher: O'Reilly Media, 2009.
3. Speech and Language Processing, Latest Edition, Daniel Jurafsky and James H. Martin Publisher: Prentice Hall, 2008 (or latest if available).

Course Name **Neural Networks and Fuzzy Logic (CS-679)**
Credit Hours **3 (2+1)**
Contact Hours **3**
Pre-requisites

Course Introduction:

This course introduces students to the fundamentals and applications of Neural Networks and Fuzzy Logic in intelligent systems. It covers the theory and design of artificial neural networks, including learning algorithms, architectures, and problem-solving capabilities. The course also explores fuzzy set theory, fuzzy inference systems, and their use in reasoning under uncertainty. Students learn how to integrate neural networks and fuzzy logic to develop intelligent systems that can model, analyze, and solve real-world problems with imprecise or complex data.

Course Learning Outcomes:

1. Explain the fundamental concepts and architectures of artificial neural networks.
2. Apply learning algorithms such as backpropagation to train neural networks for practical problems.
3. Understand fuzzy set theory, fuzzy logic principles, and fuzzy inference systems.
4. Design and implement intelligent systems that integrate neural networks and fuzzy logic.
5. Analyze and solve real-world problems using soft computing techniques and model complex data effectively.

Course Outline:

Introduction to Neural Networks, Biological Neurons and Artificial Neuron Models, Perceptron and Multi-Layer Perceptron (MLP), Activation Functions, Learning Algorithms, Backpropagation, Recurrent and Convolutional Neural Networks, Hebbian Learning, Competitive Learning, Fuzzy Set Theory, Fuzzy Relations, Membership Functions, Fuzzy Rules and Fuzzy Inference Systems, Defuzzification Methods, Neuro-Fuzzy Systems, Hybrid Intelligent Systems, Applications of Neural Networks and Fuzzy Logic in Pattern Recognition, Control Systems, and Decision Making, Case Studies and Practical Implementations.

Reference Materials:

1. Fuzzy Logic and Neural Networks for Hybrid Intelligent System Design, Oscar Castillo & Patricia Melin, Springer, 2023.
2. New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics, Oscar Castillo & Patricia Melin, Springer, 2024.
3. Neural Networks, Fuzzy Systems and Other Computational Intelligence Techniques for Advanced Process Control, Jie Zhang & Meihong Wang, MDPI Books, 2024.
4. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, Bart Kosko, 1st Edition, Prentice Hall, 1992.
5. Fuzzy Sets and Fuzzy Logic: Theory and Applications, George J. Klir & Bo Yuan, 2nd Edition, Prentice Hall, 2001.

BS - Computer Science Mathematics & Supporting Courses (12/138) 4 Courses		
Course Code	**Course Title**	**Cr. Hrs.**
CS-457	Multivariable Calculus	3
CS-356	Linear Algebra	3
CS-459	Probability & Statistics	3
CS-652	Financial Accounting	3

Course Name **Multivariable Calculus (CS-457)**
Credit Hours **3 (3+0)**
Contact Hours **3-0**
Pre-requisites

Course Introduction:
Multivariate calculus uses linear algebra to extend the important concepts of single-variable calculus to higher-dimensional settings. Topics include scalar-valued and vector-valued functions, graphs, level sets, limits and continuity; partial derivatives, gradients, tangent planes, differentiability, total derivatives, directional derivatives; paths, velocity, acceleration, arclength, curvature, vector fields, divergence, curl; extrema, Hessians, Lagrange multipliers; multiple integrals, change of variables, Jacobians; line integrals, Green's theorem; surface integrals, Stokes' theorem, and Gauss' theorem

Course objectives:
The course objective is that its successful completion should develop understanding of multivariable functions, partial differentiation and multiple integrals. The applications will be covered from several engineering problems. The other objective is to learn basic vector differential operators, gradient, divergence and curl along with their applications to calculate surface integrals, flows, flux across surfaces, Fourier series and transforms.

Course Learning Outcomes:
1. To provide students with a good understanding of the concepts and methods of multivariate calculus, described in detail in the syllabus.
2. To help the students develop the ability to solve problems using multivariate calculus.
3. To connect multivariate calculus to other fields both within and without mathematics.
4. To develop abstract and critical reasoning by studying proofs as applied to multivariate calculus

Course Outline:
Parametric equations ,Polar coordinates ,Polar coordinates ,Vectors, dot product ,dot product, determinant, cross product ,Equations of lines and planes, quadrics ,vector functions, derivatives ,Functions of several variables, limits, continuity ,Partial derivatives, tangent planes ,Chain rule ,Gradient, directional derivative, Max-min problems ,Lagrange mulitpliers ,Double integrals ,Double integrals in polar coordinates and applications ,Surface area ,Change of variables in double integrals ,Triple integrals ,Triple integrals in spherical coordinates ,Vector fields and line integrals ,Gradient fields, fundamental theorem for line integrals ,Green's theorem ,Veteran's day,Curl and divergence, Surface area ,Surface integrals, Stokes theorem ,The divergence theorem

Reference Materials:
1. James Stewart, Multivariable Calculus, Early transcendentals for UC Berkeley, 7th edition (January 1, 2011). (http://calstudentstore.berkeley.edu/courselisting/index/loadMaterials)
2. Vector Calculus, fourth edition, by Susan J. Colley, 4th Edition. (September 28, 2011)

Course Name: **Linear Algebra (CS-356)**
Credit Hours: **3 (3+0)**
Contact Hours: **3**
Pre-requisites: **Calculus and Analytic Geometry**
Course Introduction:
To provide fundamentals of solution for system of linear equations, operations on system of equations, matrix properties, solutions and study of their properties.

Course objectives:
The main aim of this course is to give students some basic ideas of calculus, which is the mathematics of motion. The purpose is not just making the students learn these ideas but to enable them apply these ideas to solve problems of practical nature. It will help the students of engineering, computer science and bioinformatics to understand and solve the problems of mathematical and logical nature in other courses of these disciplines.

Course Learning Outcomes:

1. Understand and manipulate vector and matrix operations.
2. Solve systems of linear equations using various methods.
3. Grasp the concepts of vector spaces, subspaces, bases, and dimensions.
4. Apply eigenvalues and eigenvectors in practical scenarios, especially in AI and machine learning models.
5. Use linear transformations to model real-world problems in AI.
6. Implement linear algebra operations using computational tools such as Python and MATLAB.

Course Outline:

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

Reference Materials:

1. Elementary Linear Algebra by Howard Anton, 10th edition. (January 1, 2010)
2. Linear Algebra and its Applications by Gibert Strang, 4th edition. (January 1, 2006)

Course Name: **Probability and Statistics (CS-459)**
Credit Hours: **3 (3+0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

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

Course objectives:

To provide students a deeper understanding about the statistical data its types, collection, interpretation and analysis of data. Learn and use the concepts of theory of Probability. To provide students, the knowledge of Discrete and Continuous Probability distribution and their applications in computer engineering. To enable the students to learn and apply the tools for curve fitting via Linear Regression and Correlation.

Course Learning Outcomes:

1. Understand and apply the basic principles of probability theory.
2. Analyze data using statistical methods and draw valid conclusions.
3. Use probability models to describe random processes and uncertainty.
4. Apply statistical techniques in AI and machine learning problems.
5. Use software tools for statistical analysis and data visualization.

Course Outline:

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

Reference Materials:

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)
2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, (4th edition. January 1, 2012)
3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, 3rd edition.(August 26, 2008)

Course Name **Financial Accounting (CS-558)**
Credit Hours **3 (3+0)**
Contact Hours **3-0**
Pre-requisites **None**

Course Introduction

Financial Accounting introduces students to the fundamental principles, concepts, and techniques used to measure, record, and report financial information for business decision-making. The course explains how organizations prepare financial statements, analyze transactions, and communicate financial performance to internal and external stakeholders. Students learn the role of accounting in business, the structure of financial reports, and the regulatory and ethical frameworks that guide professional practice.

Course Objective

The objective of this course is to develop students' ability to understand, apply, and interpret financial accounting information. The course aims to equip learners with practical skills in recording business transactions, preparing financial statements, and evaluating an organization's financial health. It also builds a foundational understanding of accounting standards, ethical considerations, and the role of financial information in managerial and investment decisions.

Course Learning Outcomes

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

1. Explain the purpose, scope, and principles of financial accounting.
2. Record business transactions using the double-entry system and prepare basic journals and ledgers.
3. Prepare core financial statements, including the income statement, balance sheet, and statement of cash flows.
4. Analyze financial statements to assess profitability, liquidity, and solvency.
5. Understand the role of accounting standards, regulatory frameworks, and ethical practices in financial reporting.

Course Outline

Topics include the accounting cycle, source documents, double-entry bookkeeping, journals, ledgers, trial balances, and the preparation of financial statements such as the income statement, balance sheet, and cash flow statement. Students will explore accounting concepts such as accruals, depreciation, inventory valuation, adjustments, internal controls, and financial reporting standards. The course also introduces financial analysis tools, ratio analysis, budgeting, and cost considerations relevant to software development, IT projects, and technology startups. Emphasis is placed on interpreting financial information for decision-making in computing environments.

Reference Books

1. Financial Accounting by Jerry J. Weygandt, Paul D. Kimmel, and Donald E. Kieso, 11th Edition, Wiley, 2020.
2. Accounting Principles by Jerry J. Weygandt, Paul D. Kimmel, and Donald E. Kieso, 13th Edition, Wiley, 2018.
3. Financial Accounting: Tools for Business Decision Making by Paul D. Kimmel, Jerry J. Weygandt, and Donald E. Kieso, 9th Edition, Wiley, 2019.
4. Fundamental Financial Accounting Concepts by Thomas P. Edmonds, Christopher Edmonds, Philip Olds, and Mark Maher, 11th Edition, McGraw-Hill Education, 2020.

BS – Software Engineering General Education Requirement (39/138) 15 Courses		
Sr. No.	**Course Title**	**Cr. Hrs.**
1.	Introduction to Information & Communication Technologies	3
2.	Functional English	3
3.	Expository Writing	3
4.	Quantitative Reasoning – 1 (Discrete Structures)	3
5.	Quantitative Reasoning – 2 (Calculus and Analytic Geometry)	3
6.	Islamic Studies	2
7.	Ideology and Constitution of Pakistan	2
8.	Natural Science (**Applied Physics**)	3
9.	Arts & Humanities (**Professional Practices**)	3
10.	Civics and Community Engagement	2
11.	Entrepreneurship	3
12.	Social Sciences (**URDU**)	2
13.	Fehm e Quran	2
14.	Pakistan Studies	2
15.	Internship	3

Course Name: Introduction to Information and Communication Technologies (CS-353)

Credit Hours: 3 (2+1)

Contact Hours: 3-0

Pre-requisites: None

Course Introduction:

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

Course Learning Outcomes:

1. Understand basics of computing technology
2. Do number systems conversions and arithmetic
3. Have knowledge of types of software
4. Have knowledge of computing related technologies

Course Outline:

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

Reference Materials:

1. Understanding Computers: Today and Tomorrow, Charles S. Parker, 16th edition. (February 18, 2016)
2. An introduction to automatic digital computers by Livesley, Robert Kenneth, 1st edition. (February 23, 2017)
3. Exploring four decades of research in Computers & Education by Zawacki-Richter, Olaf, and Colin Latchem, (2018).
4. Computer fundamentals by Sinha, Pradeep K., and Priti Sinha, 8th edition. (January 1, 2003)
5. Computer fundamentals by Goel, Anita, 1st edition. (December 1, 2010)

Course Name: **Functional English (CS-359)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

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

Course Learning Outcomes:

1. Enhance their understanding and application of English grammar.
2. Develop a strong vocabulary relevant to academic and professional use.
3. Improve their reading comprehension and critical analysis of texts.
4. Strengthen their writing skills, focusing on academic and technical writing.
5. Gain confidence in oral communication, including presentations and discussions.
6. Learn techniques for effective communication in professional environments.
7. Writing academic and technical reports

Course Outline:

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

Reference Materials:

1. College Writing Skills with Readings, by John Langan, McGraw-Hill, 11th edition.(January 1, 2022)
2. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, (2017)

Course Name: **Expository Writing (CS-454)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **Functional English**

Course Introduction:
Expository Writing is built on the premise that the craft of good explanatory and descriptive writing is inseparable from the ability to read widely, critically and efficiently. And so, even as English II consolidates the groundwork done in the English I module it advances in two skill areas, in particular. Firstly, it seeks to enable students to skillfully read pieces in diverse genres—including texts in English relevant to their respective fields of study. Secondly, it looks to empower students to produce a variety of argumentative and explanatory texts. The course also expands on students' grammar and vocabulary skills, as building blocks of written communication, relevant to the academic and prospective professional needs of learners. As in English I, communication proficiency goals are critically integrated with the broader educational, social and professional concerns within the Pakistani postcolonial setting.

Course objectives
The course aims to upgrade the spoken and written English of the Electronic Engineering students from an intermediate to advanced level, and to help them overcome their fear of public speech. It will also focus on introducing the basic research skills and writing skills affiliated to research, to help them in writing research papers for the contemporary Engineering courses. It also aims to inculcate confidence and to groom their personality so that they can aim at executive level jobs. The career focus of this course is to build the combination of language and interpersonal skills needed to work independently, to lead teams effectively, and to become customer focused and result driven in their approach. Special emphasis is also laid on developing individual and group effort through virtual and real life training, presentations, projects and research, especially towards developing their leadership qualities and the ability, to improve team functioning and team output.

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

Course Learning Outcomes:

1. Understand the fundamental principles of communication.
2. Develop effective verbal and non-verbal communication skills.
3. Enhance public speaking and presentation abilities.
4. Learn techniques for professional communication, including emails and reports.
5. Build confidence in delivering technical presentations.
6. Improve their skills in participating in group discussions, debates, and meetings.

Reference Materials:

1. Practical Business English, Collen Vawdrey, (1993).
2. Effective Communication Skills: The Foundations for Change, John Nielsen, (2008.)
3. College Writing Skills with Readings, by John Langan, 11th edition. (January 1, 2022)
4. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute (July 15, 2011)

Course Name: **Discrete Structures (CS-358)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

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

Course Learning Outcomes:

1. Understand the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs and Trees etc.
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.
3. Apply discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.
4. Differentiate various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular

Course Outline:

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

Reference Materials:

1. Discrete Mathematics and Its Applications by Kenneth H. Rosen, 8th edition. (July 9, 2018)
2. Discrete Mathematics with Applications by Susanna S. Epp, 4th edition. (August 4, 2010)
3. Discrete Mathematics by Richard Johnson Baugh, 8th edition. (March 6, 2017)
4. Discrete Mathematical Structures by Kolman, Busby & Ross, 4th edition. (November 23, 1999)
5. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi, 5th edition. (July 27, 2003)
6. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred, (2007).

Course Name: **Calculus and Analytical Geometry (CS-355)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**

Course Introduction:

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

Course Learning Outcomes:

1. Understand the fundamental concepts of differential and integral calculus.
2. Apply calculus to solve problems in optimization, curve sketching, and area calculation.
3. Analyze and interpret conic sections using analytical geometry.

4. Utilize calculus concepts in AI applications such as machine learning algorithms, neural networks, and robotics.
5. Develop problem-solving skills relevant to AI technologies through mathematical modeling.

Course Outline:

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

Reference Materials:

1. Calculus and Analytic Geometry by Kenneth W. Thomas. 9th edition. (August 14, 1995)
2. Calculus by Stewart, James. 9th edition. (April 30, 2020)
3. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole, 6th edition. (September 25, 1996)

Course Name: **Islamic Studies (CS-361)**
Credit Hours: **2 (2-0)**
Contact Hours: **2-0**
Pre-requisites: **None**

Course Introduction:

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

Course Learning Outcomes:

1. To further enhance the knowledge of Islam.
2. To understand the basic concept of Islam and Quran Pak.
3. To understand the concept of Haqooq ul ibad in the light of Quran.
4. To know the importance of Islamic concept about other religions.

Course Outline:

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

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

1. Introduction to Islam by Dr Hamidullah, Papular Library Publishers Lahore(2020)
2. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI, (1993).
3. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services, (1982).

Course Name: **Ideology and Constitution of Pakistan (CS-362)**
Credit Hours: **2 (2-0)**
Contact Hours: **2-0**
Pre-requisites: **None**

Course Introduction:

Pakistan studies is an important course at this university in which students' study about their motherland. The following are the specific objective of the course:

- to develop vision of Historical Perspective, Government, Politics, Contemporary Pakistan, ideological background of Pakistan.
- To study the process of governance, national development, issues arising in the modern age and posing challenges to Pakistan.

Course Learning Outcomes:

1. To educate students about the history of Pakistan
2. To educate student about the various pillar of the state
3. To educate student Government and politics

Course Outline:

Historical background of Pakistan: Muslim society in Indo-Pakistan, the movement led by the societies, the downfall of Islamic society, the establishment of British Raj- Causes and consequences. Political evolution of Muslims in the twentieth century: Sir Syed Ahmed Khan; Muslim League; Nehru; Allama Iqbal: Independence Movement; Lahore Resolution; Pakistan culture and society, Constitutional and Administrative issues, Pakistan and its geo- political dimension, Pakistan and International Affairs, Pakistan and the challenges ahead.

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

1. The Emergence of Pakistan, Chaudary M., (2009 Edition)
2. The making of Pakistan, Aziz. (2013 Edition)
3. A Short History of Pakistan, I. H. Qureshi, ed., Karachi, 2nd Edition (2000)

Course Name: **Fehm e Quran (CS-364)**
Credit Hours: **2 (2+0)**
Contact Hours: **2-0**
Pre-requisites: **None**

Course Introduction:

A "Fehm-e-Quran" course aims to provide a deep understanding of the Holy Quran by going beyond translation to teach its meaning, themes, and context through the study of Arabic grammar and Tafsir.

Course objectives

To provide students with a practical and direct understanding of the Quran, focusing on linguistic comprehension of verses and building the ability to comprehend the text independently. The course aims for students to understand a significant portion of the Quranic text and its basic and advanced linguistic components.

Course Learning Outcomes:

1. Interpret Quranic verses using various methodologies of Tafsir
2. Discuss the implications of Quranic teachings in contemporary life
3. Adopt and practice the teachings of the Quran in their daily life.

Course Outline:

This course mainly focuses on basic words, phrases, and sentences without verbs, and progresses to sentences with present tense verbs, past tense verbs and translating verses from the Quran.

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

Muallim ul Quran (Volume 1, 2 & 3) by Dr Ubaid ur Rahman

Course Name: **Pakistan Studies (CS-363)**
Credit Hours: **2 (2+0)**
Contact Hours: **2-0**
Pre-requisites: **None**

- As per University of Karachi Policy.

Course Name: **Introduction to Management (CS-462)**
Credit Hours: **2 (2-0)**
Contact Hours: **2-0**
Pre-requisites: **None**

Course Introduction:

Introduction to Management familiarizes students with core management principles, functions, and practices, including planning, organizing, leading, and controlling, while developing skills in decision-making, leadership, and teamwork in organizational contexts.

Course Learning Outcomes:

1. Explain the fundamental principles and functions of management.
2. Apply planning, organizing, leading, and controlling techniques in organizational contexts.
3. Demonstrate effective decision-making and problem-solving skills.
4. Understand leadership, motivation, and communication in managing teams.
5. Evaluate real-world organizational situations and recommend effective management strategies.

Course Outline:

Introduction to Management, Management Theories and Evolution, Challenges of Management, Pioneering Ideas in Management, Social Responsibility & Ethics, Understanding Environment & Culture, Managerial Decision Making, Organizational Goals & Plans, Planning, Organizing, Leading, Controlling, Strategic Management, Organization Structure & Design, Managing Human Resources, Managing Change & Innovation, Motivation, Leadership, Managerial Communication, Groups & Teamwork, Performance Management, Contemporary Issues in Management, Small Business Management, Case Studies and Practical Applications

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

1. Management: Tasks, Responsibilities, Practices, by Peter F. Drucker, 1st Edition, Harper & Row, 1974.
2. Principles of Management, by Charles W. L. Hill & Steven McShane, 8th Edition, McGraw-Hill Education, 2019.
3. Management: Leading & Collaborating in a Competitive World, by Thomas S. Bateman & Scott A. Snell, 14th Edition, McGraw-Hill Education, 2021.
4. Fundamentals of Management, by Stephen P. Robbins & Mary Coulter, 10th Edition, Pearson, 2018.
5. Management: A Contemporary Approach, by David A. Whetten & Kim S. Cameron, 10th Edition, Pearson, 2020.

Course Name: **Applied Physics (CS-357)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites: **None**

Course Introduction:

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

Course Learning Outcomes:

Applied Physics course, you will be able to:

1. apply core physics principles to real-world problems
2. conduct experiments that demonstrate these concepts.
3. enhance your analytical skills and effectively communicate scientific ideas.

Course Outline:

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

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

1. Fundamentals of Physics (Extended) by Resnick and Walker, 12th edition. (2021)
2. Physics for Computer Science Students by Narciso Garcia, Arthur Damask, Steven Schwarz, 2nd edition.(January 9, 1998)

Course Name: **Professional Practices (CS-462)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
Pre-requisites: **None**

Course Introduction:

A Computing graduate as professional has some responsibilities with respect to the society. This course develops student understanding about historical, social, economic, ethical, and professional issues related to the discipline of Computing. It identifies key sources for information and opinion about professionalism and ethics. Students analyze, evaluate, and assess ethical and professional computing case studies.

Course Learning Outcomes:

1. To develop key skills in ethical decision-making and effective communication in a professional environment.
2. To learn how to navigate workplace dynamics and enhance your collaboration and leadership abilities

Course Outline:

Historical, social, and economic context of Computing (software engineering, Computer Science, Information Technology); Definitions of Computing (software engineering, Computer Science, Information Technology) subject areas and professional activities; professional societies; professional ethics; professional competency and life-long learning; uses, misuses, and risks of software; information security and privacy; business practices and the economics of software; intellectual property and software law (cyber law); social responsibilities, software related contracts, Software house organization. Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

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

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, 3rd edition. (21 September 2000)
2. Computer Ethics by Deborah G. Johnson, Pearson, 4th edition. (December 24, 2008)
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet by Sara Baase, 3rd edition. (January 6, 2008)
4. Applied Professional Ethics by Gregory R. Beabout (December 9, 1993)

Course Name: Entrepreneurship (CS-653)

Credit Hours: 3 (3-0)

Contact Hours: 3-0

Pre-requisites: None

Course Introduction:

This course aims to provide students with a basic understanding of the principles and practices of entrepreneurship. It also aims to cultivate an entrepreneurial mindset and equip students with the knowledge and skills necessary to start, manage, and grow their own ventures. Learning components of ‘Financial Literacy’ and ‘Export Management’ are also included in the learning outcomes of the course to provide students with a holistic understanding of business in priority areas of the economy

Course objectives:

Entrepreneurship is an interdisciplinary course designed to teach students how to think and act entrepreneurially. Students learn how to start-ups and operate a business. The course will build cross-curricular academic skills, by integrating inquiry-based learning and business tools that will enable students to analyze, create, develop and pilot small businesses. Concepts and skills are reinforced by a strong emphasis on hands-on experiences. Applications to society, individuals, and the utilization of technology are included. This course includes a broad series of lessons and activities that offer a variety of modalities for ultimate student engagement and content retention.

Course Learning Outcomes:

1. Explain the fundamental concepts and principles of entrepreneurship.
2. Identify and evaluate business opportunities and develop innovative ideas.
3. Prepare basic business plans, including financial, marketing, and operational strategies.
4. Demonstrate knowledge of legal, ethical, and social responsibilities in business.
5. Apply entrepreneurial thinking and problem-solving skills to start or manage small businesses successfully.

Course Outline:
Introduction to Entrepreneurship, Perspectives, Benefits and Drawbacks. Types of Entrepreneurs. Cultural Diversity and Entrepreneurship. Contexts of entrepreneurial activity. Characteristics of Entrepreneur: Creativity, Professionalism, Risk-taking, Planning etc. Entrepreneurial functions, Managerial Functions: Planning, Organizing, Staffing, Directing, Controlling. Generating Ideas, Creativity and Innovation, Conceptual framework and Techniques. Feasibility Analysis: Product, Service, Marketing, Financial. Development of Business Plan, Business models, Guidelines, General Categories. Business Marketing, Strategies for Marketing. Creating Values for Customers. Strategic Management Process, Steps of Strategic Management Process. Leadership and Social Entrepreneurship. Intrapreneurs, History of Intrapreneurship, Benefits of Intrapreneurship, Characteristics of an Intrapreneur. Comparison of Managers, Intrapreneurs and Entrepreneurs. Ethical and Legal issues for Permits and Licensing. Intellectual Property Rights. Finance and Funding, Sources of Personal financing, Debit and Equity Financing. Franchising. The Process of Growth, Reasons, Strategies, Readiness, and Challenges of Entrepreneurship.

Reference Materials:

1. Entrepreneur: What It Means to Be One and How to Get Started". December 2022.
2. Martiarena, A. (2013). What's so entrepreneurial about intrapreneurs? Small Business Economics, 40(1), 27-39. doi:10.1007/s11187-011-9348-1 Mason, C., & Brown, R. (2014). Entrepreneurial ecosystems and growth oriented entrepreneurship. Retrieved from The Hague, Netherlands: http://www.oecd.org/cfe/leed/entrepreneurial-ecosystems.pdf
3. Matthews, C. H., & Brueggemann, R. (2015). Innovation and entrepreneurship: A competency framework. New York: Routledge.
4. Matzler, K., Veider, V., & Kathan, W. (2015). Adapting to the sharing economy. MIT Sloan Management Review, 56(2), 71-77.
5. Carree, M. A., Thurik, A. R. "The impact of entrepreneurship on economic growth" In: Audretsch, D. B., Acs, Z. J. (eds). Handbook of Entrepreneurship Research. Berlin: Springer Verlag, 2010.

Course Name: Civics and Community Engagement (CS-559)
Credit Hours: 2 (2-0)
Contact Hours: 2-0
Pre-requisites: None

Course Name: **Civics and Community Engagement (CS-559)**
Credit Hours: **2 (2+0)**
Contact Hours: **2-0**
Pre-requisites: **None**

Course Introduction:
This course aims to bring responsible citizenship and active engagement between Universities/HEIs (through their students) and local communities. The course will provide students with a foundational understanding of the principles, institutions, and processes of civic engagement in a democratic society. Moreover, the course will build the capacity of students as leaders and influencers by gaining fundamental understanding of leadership, citizenship, communication, advocacy, network building as well as having first-hand experience of community development through volunteer work**.**

Course objectives
The Civics and Community Engagement course is designed to help students understand the principles of civic responsibility, government functions, and the importance of active citizenship in fostering positive change. Through discussions, projects, and community engagement activities, students will explore the structures of government, social justice issues, and methods for impactful community involvement. The course aims to inspire students to be informed citizens, to advocate for social issues, and to contribute meaningfully to their communities, emphasizing the power of civic participation in strengthening society.

Course Learning Outcomes:

1. **Understand Civic Responsibilities**: Explain the principles of civic duty, community involvement, and the roles of government and individuals in society.
2. **Analyze Social Issues**: Identify and analyze pressing social, political, and economic issues within local, national, and global contexts, understanding their impact on communities.
3. **Engage in Community-Based Learning**: Participate in activities that foster community engagement, applying course concepts to real-world scenarios through volunteer work, community projects, or partnerships.
4. **Develop Critical Thinking on Public Policy**: Evaluate public policies and proposals, considering their implications for social justice, equity, and community well-being.
5. **Practice Effective Communication and Advocacy**: Demonstrate skills in communicating ideas, advocating for community needs, and engaging with diverse audiences through written, oral, and digital means.
6. **Reflect on Personal and Social Impact**: Assess personal values and actions in relation to civic responsibilities, exploring how individual contributions can promote positive social change.
7. **Collaborate for Community Solutions**: Work collaboratively on community-based projects, developing strategies to address social issues and strengthen community resilience.

Course Outline:

- Introduction to Civics and Community Engagement
- Fundamentals of Civic Responsibility
- Government and Policy-Making
- Community Issues and Social Justice
- Advocacy and Activism
- Community Service and Volunteering
- Engagement through Digital Platforms
- Building and Leading Community Initiatives
- Case Studies in Civic Engagement
- Personal Civic Action Plan

Reference Materials:
As per University of Karachi policy.

Course Name: Urdu (CS-461)
Credit Hours: 2 (2-0)
Contact Hours: 2-0
Pre-requisites: None

Course Introduction:
As per guidelines of NCEAC and HEC

Course Learning Outcomes:
As per guidelines of NCEAC and HEC

Course Outline:
As per guidelines of NCEAC and HEC

Reference Materials: (or use any other standard and latest books)
As per guidelines of NCEAC and HEC

Course Name: Internship (CS-657)
Credit Hours: 3 (3+0)
Contact Hours: 3-0
Pre-requisites: None

- As per guidelines of NCEAC and HEC

Course Name: **Professional Certification (CS-654)**
Credit Hours: **3 (0+3)**
Contact Hours: **3-0**
Pre-requisites: **None**

Course Introduction:

Integration of International Professional Certifications into BS Computer Science Program.

1. To recognize selected international certifications as equivalent to **one elective (3 credit hours)** in BS Computing programs.

2. To ensure that recognition of certifications does not compromise academic integrity or NCEAC accreditation standards.

3. To enhance employability of graduates by combining formal degree education with verifiable industry credentials.

4. List of certifications recommended by PSEB – Ministry of IT & Telecom will be approved by BoS, BoF, and Academic Council and the university may add more certifications.

5. Government and other funding avenues will cover the cost of certifications for the university students and affiliated colleges.

Course Learning Outcomes:

As per guidelines of NCEAC and HEC

Course Outline:

As per guidelines of NCEAC and HEC