Bachelor of Science in Software Engineering (BSSE) Program

Software Engineering is the discipline of developing and maintaining software systems that behave reliably and efficiently, and are affordable to develop and maintain. However, more recently it has evolved in response to the increased importance of software in safety-critical applications and to the growing impact of large and expensive software systems in a wide range of situations. The following describes an overview of Software Engineering.

- To provide software development practices requires more than just the under lying principles of computer science; it offers the rigor that the engineering disciplines bring to the reliability and trust worthiness of the artefacts.
- Software Engineering is different in character from other engineering disciplines, due to both the in tangible nature of software and to the discontinuous nature of software operation.
- It seeks to integrate the science of Computer Science with the engineering principles developed for tangible and physical phenomena.

Software plays a central and underpinning role in almost all aspects of daily life: communications, government, manufacturing, banking and finance, education, transportation, entertainment, medicine, agriculture, and law. The number, size, and application domains of computer programs have grown dramatically; as a result, huge sums are being spent on software development. Most people's lives and livelihoods depend on this development's effectiveness. Software products help us to be more efficient and productive. They provide information, make us more effective problem solvers, and provide us with safer, flexible, and less confining work, entertainment, and recreation environments.
Software Engineering is the application of a systematic, disciplined, and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient, and economical software by applying the principles and practices of engineering. The department aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evolution of software product.

Proposed Curriculum for

Bachelor of Science in Software Engineering (BSSE) Program

Proposed curriculum (semester-wise course plan and contents of the courses) for BSSE 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. Mapping of the proposed curriculum with the existing and its mapping with NCEAC/HEC curriculum are given as Annexure – A and Annexure - B.

Summary Table for Total Number of courses offered and credit hours in the program – BSSE

Course Categories	Number of Courses	Credit Hours
Computing Core + Capstone Project	16	54
Domain Core	6	18
Domain Elective	4	12
Mathematics & Supporting Courses	4	12
General Education Requirement + Internship	15	39
Professional Certification	1	3
Total Number of Courses and Credit Hours	**46**	**138**

Semester-wise Course Plan

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

Semester – II		
Course Code	**Course Name**	**Credit Hrs.**
SE-352	Object Oriented Concepts & Programming	4 (3+1)
SE-354	Software Engineering Fundamentals	3 (3+0)
SE-356	Multivariable Calculus	3 (3+0)
SE-358	Applied Physics	3 (2+1)
SE-360	Probability & Statistics	3 (3+0)
SE-362	Urdu	2 (2+0)
SE-366	Fehm e Quran	2 (2+0)
Total Credits for Semester – II		**20**

Semester – III		
Course Code	**Course Name**	**Credit Hrs.**
SE-451	Data Structures and Applications	4 (3+1)
SE-453	Software Requirement Engineering	3 (3+0)
SE-455	Digital Logic Design	4 (3+1)
SE-457	Linear Algebra	3 (3+0)

SE-459	Expository Writing	3 (3+0)
SE-450	Pakistan Studies	2 (2+0)
Total Credits for Semester – III		**19**

Semester – IV		
Course Code	**Course Name**	**Credit Hrs.**
SE-452	Database Management Systems	4 (3+1)
SE-454	Software Design and Architecture	3 (3+0)
SE-456	Computer Organization & Assembly Language	3 (2+1)
SE-458	Operating Systems	4 (3+1)
SE-460	Artificial Intelligence	3 (2+1)
SE-462	Civics and Community Engagement	2 (2+0)
Total Credits for Semester – IV		**19**

Semester – V		
Course Code	**Course Name**	**Credit Hrs.**
SE-551	Software Construction and Development	3 (2+1)
SE-553	Software Project Management	3 (3+0)
SE-555	Data Communication and Networking	3 (2+1)
SE-557	Theory of Automata	3 (2+1)
SE-xxx	Domain Elective-1	3 (3+0)
SE-559	Professional Practices	3 (3+0)
Total Credits for Semester – V		**18**

Semester – VI		
Course Code	**Course Name**	**Credit Hrs.**
SE-552	Design and Analysis of Algorithms	3 (3+0)
SE-554	Software Quality Engineering, Testing and Assurance	3 (3+0)
SE-556	Information Security	3 (2+1)
SE-558	Cloud Computing	3 (2+1)
SE-xxx	Domain Elective-2	3 (2+1)
Total Credits for Semester – VI		**15**

Semester – VII		
Course Code	**Course Name**	**Credit Hrs.**
SE-xxx	Domain Elective-3 **(Topic of Current Interest)**	3 (2+1)
SE-xxx	Domain Elective-4	3 (2+1)
SE-651	Financial Accounting	3 (3+0)
SE-653	Internship	3 (3+0)
2SE-655	Final Year Project – I	3 (0+3)
Total Credits for Semester – VII		**15**

Semester – VIII		
Course Code	**Course Name**	**Credit Hrs.**
SE-666	Professional Certification	3 (3+0)
SE-652	Parallel & Distributed Computing	3 (2+1)
SE-654	Entrepreneurship	3 (3+0)
SE-656	Final Year Project – II	3 (0+3)
Total Credits for Semester – VIII		**12**

Domain elective courses to be offered from following table:

Course Code	Domain Elective 1, 2	Credit Hrs.
SE-571	HCI and Computer Graphics	3 (3+0)
SE-572	Advanced Database Management Systems	3 (2+1)
SE-573	Object Oriented Analysis and Design	3 (3+0)
SE-574	Web Technologies	3 (2+1)
SE-575	Mobile Applications Development	3 (2+1)
SE-576	Computer Architecture	3 (3+0)
SE-577	Modelling and Simulation	3 (2+1)

Out of seven domain elective courses four 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	Credit Hrs.
SE-671	Software Re-engineering	3 (3+0)
SE-672	Software Reverse Engineering and Ethical Hacking	3 (3+0)
SE-673	Remote Sensing and Geographic Information System	3 (3+0)
SE-674	Data Science	3 (2+1)
SE-675	Data Warehousing and Data Mining	3 (2+1)
SE-677	Web Engineering	3 (2+1)
SE-678	Natural Language Processing	3 (3+0)
SE-679	Neural Networks and Fuzzy Logic	3 (3+0)

Course Outlines

Computing Core (54/141) 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 & Assembly Language	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 (SE-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 (SE-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 (SE-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 (SE-455)

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 (SE-451)**
Credit Hours: **4 (3-1)**
Contact Hours: **3-3**
Pre-requisites: **Programming Fundamentals, Object Oriented Concepts & Programming**

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 (SE-556)
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 (SE-460)

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 (SE-555)

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 (SE-354)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
Pre-requisites: **None**

Course Introduction:

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 Assembly Language (SE-456)**
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 (SE-458)

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 (SE-552)
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 Wayze

BS – Software Engineering Domain Core (18/133) 6 Courses		
S#	**Course Title**	**Cr. Hrs.**
1	Software Design & Architecture	3
2	Software Construction & Development	3
3	Software Project Management	3
4	Software Quality Engineering	3
5	Software Requirement Engineering	3
6	Parallel & Distributed Computing	3

Course Name: Software Design and Architecture (SE-454)
Credit Hours: 3 (3-0)
Contact Hours: 2-3
Pre-requisites: Software Requirement Engineering

Course Introduction:

Course Learning Outcomes:

1. Understand the role of design and its major activities within the OO software development

process, with focus on the Unified process.
2. Comprehend the advantages of consistent and reliable software design.
3. Design OOD models and refine them to reflect implementation details
4. Apply and use UML to visualize and document the design of software systems.
5. Implement the design model using an object-oriented programming language.

Course Outline:

Software Design Concepts, Design principles, Object-Oriented Design with UML, System design and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Design Patterns, Exploring inheritance, Interactive systems with MVC architecture, Software reuse. Architectural design issues, Software Architecture, Architectural Structures & Styles, Architectural Patterns, Architectural & Design Qualities, Quality Tactics, Architecture documentation, Architectural Evaluation, Model driven development.

Reference Materials:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim
2. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath
3. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures by Hassan Gomaa
4. Head First Design Patterns, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates

Course Name: Software Construction and Development (SE-551)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: Software Design and Architecture, Object Oriented Programming

Course Introduction:

Course Learning Outcomes:

1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process
2. Develop Object-oriented design models and refine them to reflect implementation details
3. Evaluate different architectures for a medium size software.
4. Implement design model using an object-oriented programming language.

Course Outline:

Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life cycle model characteristics, Individual and team software process, Lehman's Laws, code salvaging, and configuration management. Martin Fowler's refactoring concepts and their application to small projects. Apply Michael Feathers' "legacy code" concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).

Reference Materials:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim
2. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath
3. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures by Hassan Gomaa
4. Head First Design Patterns, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates

Course Name: Software Project Management (SE-553)

Credit Hours: 3 (3-0)

Contact Hours: 3-0

Pre-requisites: Software Engineering Fundamentals

Course Introduction:

Course Learning Outcomes:

1. Explain principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding

2. Critically evaluate and discuss the issues around project management and its application in the real world with course participants and learners.
3. Choose project management techniques for IT projects to initiate, plan, execute and evaluate a project and work in teams to create a project plan for a project scenario that includes key tasks, critical path, dependencies and a realistic timeline.
4. Present strategies for gaining confidence in managing projects through simple project planning examples.

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
2. A Guide to the Project Management Body of Knowledge (PMBOK Guides)
3. Mastering Software Project Management: Best Practices, Tools and Techniques, Murali K. Chemuturi and Thomas M. Cagley Jr.
4. Effective Project Management: Traditional, Agile, Extreme, Robert K. Wysocki, Wiley

Course Name: Software Quality Engineering, Testing & Assurance (SE-554)
Credit Hours: 3 (3-0)
Contact Hours: 3-0
Pre-requisites: Software Engineering
Course Introduction:
Course Learning Outcomes:

1. Outline software testing and software quality assurance principles.
2. Prepare test case and test suites for completely testing all aspects of a system under test (SUT)
3. Analyze which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques.
4. Compile findings of a quality assurance cycle.

Course Outline:

Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing

Reference Materials:

1. Software Testing, A Craftsman's Approach, Paul Jorgensen, Taylor and Francis Group
2. Fundamentals of Software Testing, ISTE, Bernard Homes
3. Software Engineering by Ian Sommerville

Course Name: Software Requirements Engineering (SE-453)
Credit Hours: 3 (3-0)
Contact Hours: 3-0
Pre-requisites: Software Engineering
Course Introduction:

Course Learning Outcomes:

1. Describe the requirements engineering process
2. Effectively analyze software requirements for the development of cost-effective and efficient technical solutions.
3. Prepare both functional and non-functional requirements along with validation for a medium-size software system.
4. Document effective requirements in Software Requirements Specification (SRS) using clear, unambiguous requirements.

Course Outline:

Introduction to Requirements Engineering, Software Requirements, classification of requirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements Management Problems, Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.

Reference Materials:

1. Software Requirements, Wiegers K. &Beatty J.
2. Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick
3. Requirements Engineering and Management for Software Development Projects, Chemuturi

Course Name: Parallel and Distributed Computing (SE-652)

Credit Hours: 3 (2-1)

Contact Hours: 2-3

Pre-requisites: Object Oriented Programming, Operating Systems

Course Introduction:

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
2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier

BS – Software Engineering Domain Elective (21/133) 7 Courses

Course Code	Domain Elective 1, 2	Credit Hrs.
SE-571	HCI and Computer Graphics	3 (3+0)
SE-572	Advanced Database Management Systems	3 (2+1)
SE-573	Object Oriented Analysis and Design	3 (3+0)

SE-574	Web Technologies	3 (2+1)
SE-575	Mobile Applications Development	3 (2+1)
SE-576	Computer Architecture	3 (3+0)
SE-577	Modelling and Simulation	3 (2+1)
Course Code	**Domain Elective 3,4, 5**	**Credit Hrs.**
SE-671	Software Re-engineering	3 (3+0)
SE-672	Software Reverse Engineering and Ethical Hacking	3 (3+0)
SE-673	Remote Sensing and Geographic Information System	3 (3+0)
SE-674	Data Science	3 (2+1)
SE-675	Data Warehousing and Data Mining	3 (2+1)
SE-677	Web Engineering	3 (2+1)
SE-678	Natural Language Processing	3 (3+0)
SE-679	Neural Networks and Fuzzy Logic	3 (3+0)

Course Name: **Object Oriented Analysis & Design (SE-573)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **Programming Fundamentals**
Course Introduction:
Course Learning Outcomes:
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
2. Using UML: Software Engineering with Objects and Components by Perdita Stevens
3. Fundamental of Object-Oriented Design in UML by Meiler Page-Jones
4. The Unified Modeling Language User Guide by G. Booch, J. Rambaugh and I. Jakobson

Course Name: **Computer Architecture (SE-576)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
Pre-requisites: **Computer Organization and Assembly Language**
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:

As per guidelines of NCEAC and HEC

Course Name: **Theory of Automata (SE-557)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
Pre-requisites: **None**
Course Introduction:

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
2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich
3. An Introduction to Formal Languages and Automata, by Peter Linz, Jones & Bartlett Publishers
4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, New Age Publishers

Course Name: **HCI and Computer Graphics (SE-571)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
Pre-requisites:
Course Introduction:
Course Learning Outcomes:
Course Outline:
Reference Materials:

Course Name: **Advance Database Management Systems (SE572)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites: **Database 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: Web Engineering (SE-677)
Credit Hours: 3 (2-1)
Contact Hours: 2-3
Pre-requisites: Programming Fundamentals
Course Introduction:
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
2. Web Engineering, Emilia Mendes and Nile Mosley
3. Web Engineering: A Practitioners' Approach, Roger S. Pressman
4. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML, CSS, DOM, JavaScript
5. JavaScript: The Definitive Guide, David Flanagan

Course Name: Software Re-Engineering (SE-671)
Credit Hours: 3 (3-0)
Contact Hours: 3
Pre-requisites: Software Construction and Development
Course Introduction:
Course Learning Outcomes:

1. Explain the concepts and technique of software re- engineering.
2. Apply reengineering techniques to maintain and modify software systems
3. Analyze and understand maintenance related problems associated with object-oriented software systems.
4. Able to perform complex design reengineering and reverse engineering problems.

Course Outline:

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

Reference Materials:

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

Course Name: **Mobile Application Development (SE-575)**
Credit Hours: **3 (2-1)**
Contact Hours: **2-3**
Pre-requisites: **Object Oriented Programming**
Course Introduction:
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:

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

Course Name: **Cloud Computing (SE-558)**
Credit Hours: **3 (3-0)**
Contact Hours: **3**
Pre-requisites: **None**
Course Introduction:
Course Learning Outcomes:
Course Outline:

Reference Materials:

BS - Computer Science Mathematics & Supporting Courses (12/130) 4 Courses		
Sr. No.	**Course Title**	**Cr. Hrs.**
1	Multivariable Calculus	3
2	Linear Algebra	3
3	Probability & Statistics	3
4	Financial Accounting	3

Course Name: Multivariable Calculus (SE-356)
Credit Hours: 3 (3-0)
Contact Hours: 3
Pre-requisites: Calculus and Analytic Geometry

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:
- To provide students with a good understanding of the concepts and methods of multivariate calculus, described in detail in the syllabus.
- To help the students develop the ability to solve problems using multivariate calculus.
- To connect multivariate calculus to other fields both within and without mathematics.
- 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:
James Stewart, Multivariable Calculus, Early transcendentals for UC Berkeley, 7th edition (January 1, 2011)
(http://calstudentstore.berkeley.edu/courselisting/index/loadMaterials)
Vector Calculus, fourth edition, by Susan J. Colley, 4th Edition. (September 28, 2011)

Course Name: Linear Algebra (SE-457)
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 Learning Outcomes:

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
2. Linear Algebra and its Applications by Gibert Strang

Course Name: Probability and Statistics (SE-360)
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 Learning Outcomes:
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
2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter
3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel

Course Name: Technical and Business Writing (SE-557)
Credit Hours: 3 (3-0)
Contact Hours: 3
Pre-requisites: Functional English
Course Introduction:
Course Learning Outcomes:
Course Outline:
Reference Materials:

BS – Software Engineering General Education Requirement (39/141) 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 & Communication Technologies (SE-353)**
Credit Hours: **3 (3-0)**
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, Class2. Organizing Computer Facility, Centralized Computing Facility, Distributed Computing Facility, Decentralized Computing Facility, Input Devices. Keyboard and its Types, Terminal (Dump, Smart, Intelligent), Dedicated Data Entry, SDA (Source Data Automation), Pointing Devices, Voice Input, Output Devices. Soft- Hard Copies, Monitors and its Types, Printers and its Types, Plotters, Computer Virus and its Forms, Storage Units, Primary and Secondary Memories, RAM and its Types, Cache, Hard Disks, Working of Hard Disk, Diskettes, RAID, Optical Disk Storages (DVD, CD ROM), Magnetic Types, Backup System, Data Communications, Data Communication Model, Data Transmission, Digital and Analog Transmission, Modems, Asynchronous and Synchronous Transmission, Simplex. Half Duplex, Full Duplex Transmission, Communications, Medias (Cables, Wireless), Protocols, Network Topologies (Star, Bus, Ring), LAN, LAN, Internet, A Brief History, Birthplace of ARPA Net, Web Link, Browser, Internet Services provider and Online Services Providers, Function and Features of Browser, Search Engines, Some Common Services available on Internet.

Reference Materials:

1. Understanding Computers: Today and Tomorrow, Charles S. Parker
2. An introduction to automatic digital computers by Livesley, Robert Kenneth.
3. Exploring four decades of research in Computers & Education by Zawacki-Richter, Olaf, and Colin Latchem
4. Computer fundamentals by Sinha, Pradeep K., and Priti Sinha
5. Computer fundamentals by Goel, Anita.

Course Name: **Functional English (SE-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:

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
2. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute

Course Name: Expository Writing (SE-459)
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 (SE-357)**
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
2. Discrete Mathematics with Applications by Susanna S. Epp
3. Discrete Mathematics by Richard Johnson Baugh
4. Discrete Mathematical Structures by Kolman, Busby & Ross
5. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi
6. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred

Course Name: **Calculus and Analytical Geometry (SE-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:

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.
2. Calculus by Stewart, James.
3. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole

Course Name: Islamic Studies or Ethics (SE-363)
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
2. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI
3. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services

Course Name: Pakistan Studies (SE-450)
Credit Hours: 2 (2+0)
Contact Hours: 2-0
Pre-requisites: None

- As per University of Karachi Policy.

Course Name: Fehm e Quran (SE-366)
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: Ideology and Constitution of Pakistan (SE-361)
Credit Hours: 2 (2-0)
Contact Hours: 2-0
Pre-requisites: None

Course Introduction:
F 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., 1967
2. The making of Pakistan, Aziz. 1976
3. A Short History of Pakistan, I. H. Qureshi, ed., Karachi, 1988

Course Name: Applied Physics (SE-358)
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:

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
2. Physics for Computer Science Students by Narciso Garcia, Arthur Damask, Steven Schwarz

Course Name: Professional Practices (SE-559)

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:

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
2. Computer Ethics by Deborah G. Johnson, Pearson
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet by Sara Baase
4. Applied Professional Ethics by Gregory R. Beabout

Course Name: Entrepreneurship (SE-654)
Credit Hours: 3 (2-0)
Contact Hours: 3-0
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: Civics and Community Engagement (SE-462)
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: Urdu (SE-362)
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 (SE-653)
Credit Hours: 3 (3+0)
Contact Hours: 3-0
Pre-requisites: None

- As per guidelines of NCEAC and HEC

Course Name: Professional Certification (CS-666)
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

Course Name: **Financial Accounting (SE-651)**
Credit Hours: **3 (3-0)**
Contact Hours: **3-0**
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

Annexure – A

The generic structure for computing degree program given before is mapped with the BSSE program in the following tables.

Generic Structure for Computing Disciplines:

Course Categories	Number of Courses	Credit Hours
Computing Core + Capstone Project	16	54
Domain Core	6	18
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**	**141**