1.1 Programming Fundamentals

CSC-101 **Programming Fundamentals**	4(3-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand basic problem solving steps and logic constructs	C2 (Understand)
CLO-2	Apply basic programing concepts	C3 (Apply)
CLO-3	Design and implement algorithms to solve real world Problems	C3 (Solve)

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.

Sample Lab Experiments

Programming Fundamentals is a cornerstone course for beginners in computer science, focusing on foundational concepts and practical applications. Sample lab experiments for this course are designed to reinforce theoretical knowledge through hands-on practice. These experiments typically include basic input/output operations, working with data types and variables, implementing conditional statements and loops, and performing arithmetic and logical operations. Students often create programs to solve real-world problems, such as calculating grades, managing simple inventories, or building basic calculators. Advanced tasks may involve arrays, functions, and debugging simple algorithms to improve problem-solving skills. By engaging in these experiments, students gain a solid understanding of coding principles and prepare for more advanced programming challenges.

Recommended Books

1. Starting out with Programming Logic & Degins, 4th Edition, Tony Gaddis,
2. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie
3. Object Oriented Programming in C++ by Robert Lafore
4. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel
5. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

1.2 Discrete structures

CSC-110 **Discrete structures**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs and Trees etc.	C2 (Understand)
CLO-2	Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.	C3 (Apply)
CLO-3	Apply discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.	C3 (Apply)
CLO-4	Differentiate various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular	C4 (Differentiate)

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.

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

1.3 Calculus and Analytical Geometry

MTH-101 **Calculus and Analytical Geometry**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand fundamental concepts of calculus, including limits, continuity, and differentiability.	C2 (Understand)
CLO-2	Apply derivatives to solve problems involving rates of change, optimization, and motion analysis.	C3 (Apply)
CLO-3	Evaluate definite and indefinite integrals and compute areas, volumes, and solve real-world problems.	C3 (Apply)
CLO-4	Analyze and interpret the geometry of curves and conic sections using calculus tools.	C4 (Analyze)
CLO-5	Design and develop mathematical models for physical systems using calculus and geometry.	C5 (Create)
Course Outline		

Introduction to calculus, Numbers, Properties of Real number system and Complex numbers. Intervals, Types of Intervals, Inequalities, Properties of Inequalities. Absolute Values Coordinate Plane & Graph. 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.
Recommended Books
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

1.4 Application of Information & Communication Technologies

CSC-100 **Application of Information & Communication Technologies**	3(2-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** General Education
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Describe the history, basic concepts and components of computers.	C2 (Understand)
CLO-2	Explain the fundamental concepts related to software, databases and information systems.	C3 (Apply)
CLO-3	Discuss the key concepts of networks and security.	C3 (Apply)
CLO-4	Describe fundamental concepts of variables, conditional and repetitive structures, functions and user interface design in python.	C4 (Analyze)
CLO-5	Solve problems with the help of flowcharts and other tools.	C5 (Evaluate)

CLO-6	Apply computer programming concepts to write entry-level program in python.	C6 (Create)

Course Outline

Introduction, ICT: Computer and Communication Technology, Introduction to Computer System, Analog and Digital Computers Different types of computers (microcomputers, minicomputers, mainframes, supercomputers, etc.)

Exploring the Computer and their Uses Basic operations: Input, Processing, Output, Storage Basic components: Hardware, Software (Application and system), Data, Users Input Data through Keyboard and Mouse Alternate devices for input data i.e Scanner, OCR Camera and Barcode etc Output video, Sound, Monitor Various types of Printers, plotters etc Transforming data into information. How computers represent and process data. Modern CPUs Types of storage devices Measuring and Improving drive performance ASCII and EBCDIC Codes Number conversions Types of Operating Systems, User Interface of OS Enhancing an OS with utility software Survey of PC and Network OS, Networking basics, Uses of networks, Common types of networks, How networks are structured, Network media, Network Hardware, Data communication on telephone lines (usage of modems), Introduction to TCP/ IP,

Using digital data connections, Broadband connections, DSL, Cable Modem, and ATM connections, Wireless networks, Wireless 802.11, Wi-Max, creating computer programs, what is a computer program? Web development, Introduction of Databases, Database Management System, Introduction to Data-warehousing, Survey of database Systems, Introduction to data mining, Internet and the world, Email and other Internet services MS Access Data Type, Sheet, Data input, Form, Query , Report , SQL Statement, Importance of security, Understanding the need for security Measures, Taking Protective measures, Artificial Intelligence, Focus on User Interface/ Design, Pervasive Computing, wireless sensor networks RFIDs, Collaborative computing Grid, Cloud computing, Next-generation Networking (NGNs), ICT for development, Review of Course and Group Presentations and discussions, Internet and the world, Email and other Internet services, What is a computer program?, Web development

Sample Lab Experiments

The **Application of Information & Communication Technologies (ICT) with Python** focuses on equipping students with hands-on skills to solve practical problems using Python programming. Sample lab experiments are designed to integrate Python into ICT applications, such as automating file handling tasks, creating data visualization dashboards, and performing web scraping for information retrieval. Labs often involve writing Python scripts to manipulate data structures, process real-time data from APIs, and manage databases. Students also explore networking concepts by building simple client-server applications and understanding data encryption techniques. These experiments not only enhance programming skills but also demonstrate how Python can be effectively used in ICT fields like data analysis, automation, and system integration.

Recommended Books

1. Introduction to computers, Peter Norton, 7thEdition
2. Computer Science An Overview, J. Glenn Brookshear 13th Edition.

1.5 Functional English

ENG-102 **Functional English**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** General Education
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Develop an ability to comprehend writing and communicating strategies	C2 (Understand)
CLO-2	Adapt and formulate different writing skills for formal and informal situations	C3 (Apply)
CLO-3	Apply and produce documents for exchanging information in formal and informal situations (including business writing to contribute towards professional challenges etc)	C4 (Analyze)

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.

Recommended Books

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748
3. 3. College Writing Skills with Readings, by John Langan, McGraw-Hill, 5th Edition.

4. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

1.6 Pre-Calculus-1

MTH-001 **Pre-Calculus-1**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Develop an ability to comprehend writing and communicating strategies	C2 (Understand)
CLO-2	Adapt and formulate different writing skills for formal and informal situations	C3 (Apply)
CLO-3	Apply and produce documents for exchanging information in formal and informal situations (including business writing to contribute towards professional challenges etc)	C4 (Analyze)

Course Outline

Defining Set, various types of set representation and operations, Relation and function, Graphical transformation of one- and two-dimensional functions, Properties of functions composition and inverses of functions domain and range of the functions, Maximum and minimum values of functions, Increasing and decreasing functions, zeros and intercept of functions, piecewise functions, continuity, and Discontinuity of functions, Polynomial long division and Synthetic division, Solution of rational functions, Absolute valued function, properties of absolute valued functions , Asymptotes (Horizontal, vertical and oblique), Exponential functions and their properties, Logs functions and their properties, Systems of Two Equations and Two Unknowns, Systems of Three Equations and Three Unknowns, Matrix Algebra (Add, subtract and multiply matrices), Row Operations and Row Echelon Forms, Augmented Matrices, Determinant of Matrices ($2 \rightarrow 2$ and higher order matrices). Cramer's Rule, Inverse Matrices, Series and Sequences, Trigonometry, Angles in Radians and Degrees, Right Triangle Trigonometry. Law of Cosines & Sine. Area of Triangle, Graphs of Other

Trigonometric Functions, Graphs of Inverse Trigonometric Functions, Basic Trigonometric Identities (Pythagorean, Sum and Difference, Double, Half, and Power Reducing), Trigonometric Equations, General Form of a Conic, Parabolas

Circles, Ellipses, Hyperbolas, Degenerate Conics, Polar, and Parametric Equations Polar and Rectangular Coordinates.
Recommended Books
1. Textbook of Algebra and Trigonometry Class XI is published by Punjab Textbook Board (Lahore, Pakistan. 2. Calculus and Analytic Geometry, MATHEMATICS 12 (Mathematics FSc Part 2 or HSSC-Il), Punjab Text Book Board Lahore, Pakistan 3. Gilbert, S. S., B. C. Andy and B. Andrew, B. 2005. Linear Algebra and Its Applications. 4th Ed. Thomson Brooks/Cole, Belmont, CA, USA. 4. Chung, S. K. 2014. Understanding basic calculus. Create Space Independent Publishing Platform,

2.1 Object Oriented Programming

(CSC-102) **Object Oriented Programming**	4(3-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand principles of object oriented paradigm.	C-2
CLO-2	Identify the objects & their relationships to build object oriented solution	C-3
CLO-3	Model a solution for a given problem using object oriented principles	C-3
CLO-4	Examine an object oriented solution	C-4

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.
Sample Lab Experiments
This course covers a wide range of concepts in C++ programming, starting with structures, where students learn how to define structures, declare structure variables, initialize them, and access members, including nested structures. The course then delves into functions, teaching the declaration, calling, and definition of functions, passing arguments (both constants and variables), returning values, and passing structures or pointers as arguments. Object-oriented programming (OOP) concepts are introduced, including objects, classes, encapsulation, abstraction, information hiding, and access specifiers. Students learn how to work with arrays of objects, passing/returning objects, destructors, static class data, constant member functions, and constant objects. The course explores inheritance, covering derived and base classes, constructors, protected specifiers, overriding, and various inheritance levels such as single, multilevel, and multiple inheritance. Operator overloading, polymorphism (virtual and inline functions, static functions, late and early binding), memory management using new and delete keywords, and template functions (class templates) are also studied. Exception handling, including syntax for simple and multiple exceptions and exceptions with arguments, is covered, along with the Standard Template Library (STL), including function templates, class templates, vectors, deques, iterators, and function objects. Finally, the course covers file and stream handling, including streams, string I/O, character I/O, object I/O, disk I/O with member functions, and file pointers. The course concludes with project demos, a viva, and a review session.
Recommended Books
6. Object-Oriented Programming in C++, Robert Lafore; 4th Edition. 7. C++ How to Program, Harvey M. Deitel, Paul J. Deitel, Prentice Hall; 2nd Edition, 1997, ISBN: 0-13528910-6. 8. Programming and Problem Solving with C++, Nell Dale, Chip Weems, Fifth Edition, 2010, ISBN-13: 9780763771560. 9. Problem Solving and Programming Concepts, M Sprankle, Prentice Hall,

2.2 Database Systems

CSC-103 **Database Systems**	4(3-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Explain fundamental database concepts.	C-2
CLO-2	Design conceptual, logical and physical database schemas using different data models.	C-5

CLO-3	Identify functional dependencies and resolve database anomalies by normalizing database tables.	C-2
CLO-4	Use Structured Query Language (SQL) for database definition and manipulation in any DBMS	**C-4**

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 subqueries in SQL, Grouping and aggregation in SQL, concurrency control, database backup and recovery, indexes, NoSQL systems

Sample Lab Experiments

This course covers the installation and basic operations of MS SQL Server, including creating and dropping databases. Students will learn the implementation of Data Definition Language (DDL) commands, such as creating, altering, and dropping tables. The course also covers Data Manipulation Language (DML) commands, including insert, update, delete, select, and truncate, with suitable examples. Various types of SQL functions are explored, including number, aggregate, character, conversion, and date functions. Students will also learn to use different SQL operators like arithmetic, logical, comparison, and special operators, as well as set operations. The course introduces joins, including inner, outer, and natural joins, and includes the study and implementation of the GROUP BY and HAVING clauses, the ORDER BY clause, indexing, and subqueries. Additionally, students will learn to work with views, constraints, backup and recovery commands, and transactional commands like rollback, commit, and savepoint. The course also covers SQL data types, the DISTINCT, TOP, and FETCH clauses, SQL wildcard characters, auto-increment fields, administrative commands, and the implementation of SQL triggers.

Recommended Books

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

2.3 Digital Logic Design

CODE and TITLE (CSC-111) Digital Logic Design	**CREDIT & CONTACT HOURS** 3(2-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits	C-2
CLO-2	Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques	C-5
CLO-3	Apply the acquired knowledge to simulate and implement small-scale digital circuits	C-2
CLO-4	Understand the relationship between abstract logic characterizations and practical electrical implementations.	C-4

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.

Sample Lab Experiments

This course aims to provide a comprehensive understanding of digital logic principles, including logic gates and Boolean algebra, through hands-on experimentation. Students will develop proficiency in designing digital circuits, focusing on both combinational and sequential logic elements, and gain valuable skills in creating intuitive interfaces. The course also emphasizes simulation techniques to test and verify the functionality of designed circuits, bridging theoretical concepts with practical application. Students will gain practical experience in describing and simulating digital designs, while also analyzing data packets within digital system simulations, with a specific focus on the Data Link layer. Additionally, the course covers logical addressing and basic routing concepts in digital networks, utilizing CircuitVerse simulations to help students understand practical implementations. Through this, students will design and simulate basic network protocols and iteratively test and optimize digital systems for improved performance metrics.

Recommended Books

1. Digital Fundamentals by Floyd, 11/e.
2. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e
3. M. Morris Mano, Michael D. Ciletti, "Digital design", 4th edition, 2008
4. Text Book: M. Morris Mano, "Digital Logic and Computer Design", Pearson, 2011
5. Thomas L. Floyd, "Digital Fundamentals", 10th edition, Pearson Education, 2008.

2.4 Statistics & Probability

STT-101 **Statistics & Probability**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Acquire the basic knowledge of probability and probability distribution.	C1 (Remember)
CLO-2	Understand the concepts of basic techniques of measuring the uncertainty problem.	C2 (Understand)
CLO-3	Evaluate and Interpret basic descriptive statistics.	C5 (Evaluate)
CLO-4	Analyze the by using probability techniques.	C4 (Analyze)

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.
Recommended Books
1. Walpole Mayers Myers Ye, "***Probability and Statistics for Engineers and Scientists"*** 2. Sheldon M. Ross, ***" A first Course In Probability"*** 3. Neil, A. Weiss, J. Hasset "***Introductory Statistics"***, Addison-Wesley Publishers, New York. 4. Prof. Sher Muhammad Ch. ***"Introduction to Statistical Theory-I,*** Ilmi Kitab Khana, Lahore. 5. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116 6. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259 7. Prof. Sher Muhammad Ch. "Introduction to Statistical Theory-I, Ilmi Kitab Khana, Lahore.

2.5 Multivariable Calculus

CODE and TITLE MTH-102 **Multivariable Calculus**	**CREDIT & CONTACT HOURS** 3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Computing Science Supporting
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the basic concepts and know the basic techniques of differential and integral calculus of functions of several variables;	C2 (Understand)
CLO-2	Apply the theory to calculate the gradients, directional derivatives, arc length of curves, area of surfaces, and volume of solids;	C3 (Apply)
CLO-3	Solve problems involving maxima and minima, line integral and surface integral, and vector calculus;	C4 (Analyze)
Course Outline		
Functions of Several Variables and Partial Differentiation. Multiple Integrals, Line and Surface Integrals. Green's and Stoke's Theorem. Fourier Series: periodic functions, Functions of any period P-2L, Even & odd functions, Half Range expansions, Fourier Transform; Laplace Transform, Z-Transform..		

Recommended Books
1. Multivariable Calculus, 6th edition James, Stewart 2007 Cengage Learning publishers. 2. Calculus and Analytical Geometry, 6th edition. Swokowski, Olinick and Pence.1994.Thomson Learning EMEA, Ltd. 3. Multivariable Calculus, 5th edition Howard, A. Albert, H. 1995, John Wiley.

2.6 Pre- Calculus-II

MTH-002 **Pre- Calculus-II**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the basic concept of continuity and discontinuity of functions and their properties	C2 (Understand)
CLO-2	Understand the concept of derivatives, formulas, and properties related to derivative	C2 (Understand)
CLO-3	Understand the concept of derivatives, formulas, and properties related to derivative	C3 (Apply)
CLO-4	Understand the Basic definitions of definite and indefinite Integrals, Learn about the Fundamental Theorem of Calculus	C2 (Understand)

Course Outline
Understand the basic concept of Complex numbers and its arithmetic properties Learn about the idea of sequence and series, and their properties Learn about Permutations and Combinations, Basic Probability Understand the basic concept of Limits of functions, and their properties Understand the basic concept of continuity and discontinuity of functions, and their properties Understand the concept of derivatives, formulas and properties related to derivative Under the concept of Increase, Decrease, Concavity, Relative Extrema, Absolute Maxima and Minima. Understand the Basic definitions of definite and indefinite Integrals, learn about the Fundamental Theorem of Calculus Learn how to Evaluate Definite Integrals by Substitution Learn how to evaluate the integral of Logarithmic and Other Functions

Recommended Books

1. Textbook of Algebra and Trigonometry Class XI is published by Punjab Textbook Board (PTB)
2. Calculus and Analytic Geometry, MATHEMATICS 12 (Mathematics FSc Part 2 or HSSC-ll), Punjab Text Book Board Lahore
3. Mark J. Christensen, Computing for Calculus, 1st Edition, Academic Press, (1st January 1981).
   240 pages, ISBN: 9781483271088
4. Lay, L D. 2015. Probability and Statistics for Engineering and the Sciences, 9 th Ed. Cengage Learning, Boston, MA, USA.
5. Howard, Anton, Irl Bivens, Stephen Davis, Calculus, 10th Ed, 2011, John Wiley & Sons, Inc. (1318 Pages)

3.1 Data Structure

(CSC-201) **Data Structure**	4(3+3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Implement various data structures and their algorithms, and apply them in implementing simple applications	C3 (Apply)
CLO-2	Analyze simple algorithms and determine their complexities	C5 (Analyze)
CLO-3	Apply the knowledge of data structures to other application domains.	C3 (Apply)
CLO-4	Design new data structures and algorithms to solve problems	C6 (Design)

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.

Sample Lab Experiments

This course provides a comprehensive introduction to data structures, emphasizing their significance in problem-solving and algorithm implementation. Topics covered include abstraction, concrete and abstract data types, class invariants, arrays, and multi-dimensional arrays, along with complexity analysis. Students will learn about various types of linked lists, including singly, doubly, and circular lists, and their applications in polynomials and sparse matrices. The course explores stacks, recursion, sorting algorithms (merge sort, quick sort, bubble sort, and insertion sort), and queues, with applications in operating systems and message queues. Further topics include trees, binary search trees, AVL trees, multi-way trees, B-trees, heaps, Huffman codes, and graphs, including graph traversals and Dijkstra's algorithm. The course also covers hash tables, dynamic memory allocation, and advanced algorithms. Students will apply these concepts through project demonstrations and a final course review.
Recommended Books
1. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss 2. Data Structures Using C and C++, Thomas Y. Langsam, M. J. Augenstein, A. M. Tenenbaum, Prentic-Hall. 3. Data Structures and Algorithms, A. V. Aho, J. E. Hopcroft, J. D. Ullman, Addison-Wesley 4. Introduction to Algorithms, Thomas H. Cormen et al, Prentice-Hall. Lecture Notes and slides. 5. Data Structures and Algorithms in C++ by Adam Drozdek 6. Data Structures by Mark Allen Weiss (Fourth Edition),

3.2 Artificial Intelligence

(CSC-203) **Artificial Intelligence**	3(2-3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the fundamental constructs of Python programming language.	C2 (Understand)
CLO-2	Understand key concepts in the field of artificial Intelligence	C2 (Understand)
CLO-3	Implement artificial intelligence techniques and case Studies	C3 (Apply)
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, Minmax algorithm, Alpha beta pruning, Game-playing); Case Studies: General Problem Solver, Eliza, Student, Macsyma; Learning from examples; ANN and Natural Language Processing; Recent trends in AI and applications of AI algorithms. Python programming language will be used to explore and illustrate various issues and techniques in Artificial Intelligence.

Sample Lab Experiments

The course begins with an introduction to Python programming, including the installation of Python and Jupyter Notebook, followed by an overview of essential AI libraries such as NumPy, Pandas, and Matplotlib for basic data manipulation and visualization. Students then implement a basic rule-based system and experiment with early search algorithms.

In the Uninformed Search Algorithms section, students implement a range of techniques including Breadth-first Search (BFS), Depth-first Search (DFS), Depth-limited Search, Iterative deepening depth-first search, and Uniform cost search. The course also covers Bidirectional Search and explores Informed Search Algorithms, including Best First Search and A Search*. A special focus is placed on Adversarial Search, where students build a simple game-playing AI (e.g., Tic-Tac-Toe using the Minimax algorithm) and implement Alpha-Beta pruning to optimize search efficiency.

The course then shifts to Machine Learning, introducing key libraries like Scikit-Learn. Students load and preprocess datasets, then train and evaluate simple classifiers, such as k-Nearest Neighbors (k-NN), and visualize data and model performance metrics. The section on Supervised Learning covers Regression, with implementations of Linear Regression and Multiple Linear Regression, as well as evaluation metrics like Mean Squared Error (MSE) and R^2. The classification section introduces Logistic Regression, with performance evaluation, followed by the Decision Trees algorithm, where students train and visualize tree structures while exploring hyperparameter tuning and cross-validation.

In the Unsupervised Learning module, students implement k-Means clustering and evaluate clustering performance using silhouette scores. They also explore Hierarchical Clustering and visualize the results with dendrograms. The course includes an introduction to Neural Networks, where students implement both Single-layer Perceptron and Multi-layer Perceptron model

The course concludes with project presentations and a feedback session, where students showcase their final projects and reflect on the learning experience.

Recommended Books

1. Russell, S. and Norvig, P. “Artificial Intelligence. A Modern Approach”, 3rd ed, Prentice Hall, Inc., 2015.
2. Norvig, P., “Paradigms of Artificial Intelligence Programming: Case studies in Common Lisp”, Morgan Kaufman Publishers, Inc., 1992.
3. Luger, G.F. and Stubblefield, W.A., “AI algorithms, data structures, and idioms in Prolog, Lisp, and Java”, Pearson Addison-Wesley. 2009.
4. Severance, C.R., 2016. “Python for everybody: Exploring data using Python
5. ” 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

3.3 Computer Network

(CSC-204) **Computer Networks**	3(2+3) **32 Theory + 32 Lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Describe the key terminologies and technologies of computer networks	C2 (Describe)
CLO-2	Explain the services and functions provided by each layer in the Internet protocol stack	C2 (Explain)
CLO-3	Identify various internetworking devices and protocols and their functions in a networking	C4 (Identify)
CLO-4	Analyze working and performance of key technologies, algorithms and protocols	C4 (Analyze)
CLO-5	Build Computer Network on various Topologies	P3 (Build)

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 network

Sample Lab Experiments

The course starts with an introduction to basic networking commands, including Ping and Traceroute, along with their customization for gathering detailed network information. It covers basic network commands on Windows (like ipconfig) and Linux (like ifconfig). Practical skills are developed through tasks like LAN cable construction (Cat-6, Cross vs. Straight Cable), LAN cable testing, and building a physical LAN using switches. The course also introduces resource sharing (file/printer) within a Local Area Network (LAN). Students get familiar with the Cisco Packet Tracer environment, starting with basic network creation and learning to set up simple networks. They build and test different network topologies, such as bus, star, ring, mesh, and hybrid, using switches and PCs while assigning IP addresses and testing connectivity with ping. The course continues with configuring basic switch settings, including IP addressing, VLANs, and router configurations.

In-depth coverage of routing concepts includes static routing and dynamic routing protocols, such as RIP, OSPF, and EIGRP. Students learn to configure and verify these routing protocols on routers, including advanced multi-area OSPF and configuring EIGRP. The course also dives into Access Control Lists (ACLs), with practical exercises on configuring and testing standard and extended ACLs. Further topics include DHCP configuration, Network Address Translation (NAT), and WAN technologies like Point-to-Point Protocol (PPP), which is configured with authentication (PAP/CHAP). Students also explore advanced switching concepts, such as Spanning Tree Protocol (STP), RSTP, and MSTP. The course concludes with a project demo and viva, providing an opportunity to review and demonstrate the practical skills and knowledge acquired throughout the course.
Recommended Books
1. Computer Networking: A Top-Down Approach Featuring the Internet, 6th edition by James F. Kurose and Keith W. Ross Computer Networks, 5th Edition by Andrew S. Tanenbaum 2. Data and Computer Communications, 10th Edition by William Stallings 3. Data Communication and Computer Networks, 5th Edition by Behrouz A. Forouzan

3.4 Software Engineering

(CSC-205) **Software Engineering**	3(3+0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Describe various software engineering processes and Activates	C1 (Describe)
CLO-2	Apply the system modeling techniques to model a medium size software systems	C3 (Apply)
CLO-3	Apply software quality assurance and testing principles to medium size software systems	C4 (Apply)
CLO-4	Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C2 (Discuss)
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
Recommended Books
1. Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014] 2. Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015

3.5 Linear Algebra

(MTH-103) **Linear Algebra**	3(3+0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Mathematics and Science Foundation
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Solve linear systems of equations,	C2 (Understanding)
CLO-2	Comprehend vector spaces, subspaces and inner product spaces,	C2 (Understanding)
CLO-3	Understand fundamental properties of matrices including determinants, inverse matrices, matrix factorizations, eigenvalues, orthogonality and diagonalization,	C2 (Understanding)
CLO-4	Have an insight into the applicability of linear algebra,	C2 (Understanding)
CLO-5	Critically analyze and construct mathematical arguments that relate to the study of introductory linear algebra.	C3 (Application)
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
Recommended Books
1. Elementary Linear Algebra by Howard Anton 2. Linear Algebra and its Applications by Gibert Strang

4.1 Computer Organization & Assembly Language

CSC-211 **Computer Organization & Assembly Language** **Pre-requisite CSC-111**	4(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Computer Science Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Acquire the basic knowledge of computer organization computer architecture and assembly language	C2 (Understand)
CLO-2	Understand the concepts of basic computer organization, architecture, and assembly language techniques	C2 (Understand)
CLO-3	Solve the problems related to computer organization and assembly language	C3 (Apply)
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
Sample Lab Experiments
The **Computer Organization and Assembly Language** course introduces students to the fundamental concepts of computer architecture and low-level programming. Sample lab experiments are designed to provide hands-on experience with assembly language and hardware-level operations. These labs often include writing simple assembly programs to perform arithmetic operations, implement control structures (loops, conditions), and manipulate data in registers and memory. Experiments may also involve understanding instruction sets, debugging assembly code, and simulating processor instructions using tools like MASM or NASM. Advanced labs focus on interfacing with hardware, such as controlling I/O devices, working with interrupts, and optimizing code for performance. These activities aim to bridge the gap between high-level programming and underlying hardware functionality, fostering a deep understanding of how computers execute instructions.
Recommended 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, 3/E (CS:APP3e), Randal E. Bryant and David R.O' Hallaron, Carnegie Mellon University 4. Robert Britton, MIPS Assembly Language Programming, Latest Edition,

4.2 Expository Writing

ENG-201 **Expository Writing** **Pre-requisite ENG-1-2**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** General Education

After completion of this course students will be able to:		Bloom's Taxonomy Level
CLO-1	Understand proper writing conventions and grammar to write clear and concise expository essays	C2 (Understand)
CLO-2	Understand research methods to locate and integrate credible sources into writing	C2 (Understand)
CLO-3	Apply critical thinking skills to analyze and evaluate sources and arguments	C3 (Apply)
CLO-4	Analyze the rhetorical situation and use effective writing strategies and techniques to persuade readers	C4 (Analyze)

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

Recommended Books

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748
3. College Writing Skills with Readings, by John Langan, McGraw-Hill, 5th Edition.
4. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

4.3 Islamic Studies

IS-201 **Islamic Studies**	2(2-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** General Education

After completion of this course students will be able to:		Bloom's Taxonomy Level
CLO-1	To further enhance the knowledge of Islam.	C2 (Understand)
CLO-2	To understand the basic concept of Islam and the Quran Pak.	C2 (Understand)
CLO-3	To understand the concept of Haqooq ul Ibad in the light of the Quran.	C3 (Apply)
CLO-4	To know the importance of the Islamic concept about other religions.	C4 (Analyze)

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.

Recommended 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

4.4 Web Technologies

CSC-251 Web Technologies	3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Domain Elective

After completion of this course students will be able to:		Bloom's Taxonomy Level
CLO-1	Understanding the Web Development Lifecycle.	C2 (Understand)

CLO-2	Design and develop the front-end of web applications using modern web technologies (HTML, CSS, JavaScript).	C3 (Apply)
CLO-3	Understand frameworks and use them to develop a basic web application (e.g., React, Angular, Django).	C2 (Understand)
CLO-4	Design and implement advanced web applications, including features such as authentication, authorization, sessions, and reports.	C5 (Create)

Course Outline

Intro to Web Technologies, Client server architecture, protocols, html, CSS, JavaScript bootstrap, python, Django Introduction / Importance of the World Wide Web, Static & Dynamic Websites, Web Information Systems, Future of the Web (Cloud Computing, Semantic Web), demonstration of large-scale web systems, Web Engineering Introduction {Web Development is different, Web Development Practices. Three Tier Architecture, Components of Web Architecture, Identification of Places of Work in web architecture. Markup Languages Vs Programming Languages, HTML operations: Elements, tags and attributes, Block level vs in line level elements, paired vs unpaired elements

HTML Forms: Headings, styles, formatting, images, form links, Creating HTML pages, Html tables, lists, quotations, formatting, HTML media, Html Graphics, CSS Basics: Cascade Style Sheet, History of CSS, keywords of CSS, styling

CSS Selectors and its attributes: Simple Selectors, Combinator Selectors, Pseudo-Class Selectors, Pseudo-Element Selectors, Attribute Selectors, CSS Box Model and its attributes: Margin, Padding, CSS Positions, CSS Flexbox, Making webpages responsive, media queries, viewport, grid views, responsive templates, CSS animations shadows, gradients, transitions and paginations, XHTML on Mobile Phones, Transforming Pages Using XSLT, Introduction, Graphics Tools, Working with Adobe Photoshop CS, Image Slicing, Image Repetitions for creating effects, Home Page, Sub Pages Create Web Pages with Bootstrap, Bootstrap Grids and Classes ,Basic Structure of Bootstrap Grid Tables, Images, Buttons.

Intro To Java Script: History of JavaScript, What Is the Use of JavaScript? Examples of JavaScript Applications

JavaScript Functions: JavaScript Types, JavaScript Keywords, JavaScript is Case Sensitive, JavaScript Syntax, JavaScript Comments, JavaScript DOM: JavaScript HTML DOM, JavaScript, HTML DOM Methods, The DOM Programming Interface, JavaScript HTML DOM Document, JavaScript HTML DOM Elements, WDLC: full stack web development, Seven Phases of web development life cycle, Programming in Web Environment: Getting Started with the PHP, Static Vs Dynamic Pages, Dynamic Pages and PHP, Installation, Platforms & Web Servers, MySQL, Installation, Configuration & Access Privileges, Writing First Example of PHP, Common Errors with PHP, Variables, Constants and Data Types: Data Types, Literals and Identifiers, Data Values, Working with variables, Special thing about variables, String Concatenation, Constants, Operators Statements, Input Handling: Conditional Statements, Loops, Exercises, Logic with Loops.

Passing Variables between Pages, Including Files in PHP Pages, Creating Online Test Registered Users Report

Functions: Syntax, Passing Arguments, Scope, Including Functions, Variable Scope and Lifetime, Nested Functions and Recursion, Arrays: Simple Arrays, Initializing Arrays, Looping through Arrays, Multidimensional Arrays

Asynchronous JavaScript and XML: How AJAX is different from classical web development, AJAX impact on user experience, AJAX integration with conventional web applications, Use of AJAX to add, remove, create and stylize contents. Managing AJAX communication, Managing Data with XML and JSON, Design Pattern for AJAX {Overall UI Pattern, MVC Pattern. Handling and Avoiding Errors: Error Types in PHP, Generating PHP Errors, Exceptions, Not Meeting Conditions, Parse Errors, Sessions Database Connectivity & Querying: Relational Database, Keys, Relationships, Referential Integrity, Normalization, MySQL Database Operations {connect, select_db, query, fetch methods}.

Sample Lab Experiments

Installation of visual studio code and getting started with HTML, Structure of an HTML document, DOCTYPE declaration, HTML tags and attributes Creating a simple webpage, HTML Headings, Paragraphs, Lists (ordered, unordered), Text formatting tags (bold, italic, underline), HTML Hyperlinks, Adding images, Image attributes (alt, title),Creating tables, Table tags (tr, th, td), creating forms, Form elements (input, text area, select),Semantic tags (header, footer, article, section), Importance of semantic HTMLCSS intro, integration, Inline, internal, and external CSS, CSS syntax, Selectors (class, id, element), CSS Text styling (font-family, font-size, color), Box model (margin, padding, border), CSS Display property (block, inline, inline-block), Positioning (relative, absolute, fixed, sticky), Flexbox basics, CSS Attribute selectors, Pseudo-classes (: hover, nth-child), CSS Grid container and items, defining rows and columns, Placing items in the grid

CSS media Queries, Mobile-first Design, Frameworks (Bootstrap, Foundation), Introduction to JavaScript Variables, Data Types, and Operators, Functions and scope, events and Event Handling, JavaScript Arrow Functions, Classes, and Modules, Promises and Async/Await., JavaScript Selecting and Modifying Elements, Event Listeners, Creating and Removing Elements, Dynamic Content Updates, CSS Manipulation with JavaScript, JavaScript Event Listeners and Handlers, Event Delegation, JavaScript-based Animations, Libraries (jQuery, GSAP), Client-Server Model, HTTP/HTTPS Protocol, Request-Response Cycle, Asynchronous JavaScript (AJAX, Fetch API), Promises and async/await, Introduction to JavaScript frameworks (e.g., React, Vue), Introduction to PHP, PHP syntax and basic programming concepts, Variables, data types, and operators, Control structures (if, switch, loops), Working with Forms and Databases, Handling form data. Introduction to SQL and database operations, Connecting PHP with MySQL, Advanced PHP Sessions and cookies, File handling, Security best practices (input validation, sanitization), Building a Full-Stack Application, setting up a project, Combining HTML, CSS, JavaScript, and PHP, CRUD operations with PHP and MySQL , Deploying a web application

Recommended Books

1. The Modern Web: Multi-Device Web Development with HTML5, CSS3, and Javascript by Peter Gasston.
2. Beginning HTML, XHTML, CSS, and JavaScript by Jon Duckett.
3. Programming Professional PHP5 by Wiley.
4. Beginning PHP5, Apache, and MySQL Web Development by Wrox
5. https://www.w3schools.com

4.6 Web Technologies

CSC-301 **Operating Systems**	3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Computing Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems	C2 (Understand)
CLO-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	C5 (Evaluate)
CLO-3	Demonstrate the knowledge in applying system software and tools available in modern operating systems.	C3 (Demonstrate)

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

Sample Lab Experiments

Introduction to compiler construction and phases of compiler, How to create Virtual Machines, Installation of Ubuntu, Introduction to Ubuntu, shell, software installation – exploring the OS, Introduction to UNIX, UNIX architecture, UNIX commands, Directory listing, UNIX File Management — Creating files, editing files, deleting files, copying files, counting words of file, displaying content, UNIX File Management— listing directory, changing file path, hidden files, UNIX Directory Management — Home directory, creating directory, changing directory, removing directories, UNIX Pipes and Filters – Grep command and parameters, UNIX Pipes and Filters – Sort command and parameters, UNIX Network Communication Utilities – Ping utility, FTP utility, UNIX Network Communication Utilities – Telnet utility, traceroute etc. Synchronization Problems – Producer Consumer Problem and its solution, Synchronization Problems – Dinner Philosopher's Problem, Synchronization Problems – Reader Writer Problem
Recommended Books
1. Operating Systems Concepts, 9th edition by Abraham Silberschatz 2. Modern Operating Systems, 4th edition by Andrew S. Tanenbaum 3. Operating Systems, Internals and Design Principles, 9th edition by William StallingsWu

5.1 Technical & Business Writing

ENG-315 **Technical & Business Writing**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Supporting Course
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Comprehend technical writing, its characteristics, referencing, difference between technical & academic writing	(Level: C2)
CLO-2	Produce effective technical documents based on reader-based principles and clear writing style.	(Level: C3)
CLO-3	Present report writing skills using a standard word processing software along with a referencing tool and other interpersonal skills	(Level: A2)
Course Outline		

Overview of technical reporting, use of library and information gathering, administering questionnaires, reviewing the gathered information; Technical exposition; topical arrangement, exemplification, definition, classification and division, casual analysis, effective exposition, technical narration, description and argumentation, persuasive strategy, Organizing information and generation solution: brainstorming, organizing material, construction of the formal outline, outlining conventions, electronic communication, generation solutions. Polishing style: paragraphs, listening sentence structure, clarity, length and order, pomposity, empty words, pompous vocabulary, document design: document structure, preamble, summaries, abstracts, table of contents, footnotes, glossaries, cross-referencing, plagiarism, citation and bibliography, glossaries, index, appendices, typesetting systems, creating the professional report; elements, mechanical elements and graphical elements. Reports: Proposals, progress reports, Leaflets, brochures, handbooks, magazines articles, research papers, feasibility reports, project reports, technical research reports, manuals and documentation, thesis. Electronic documents, Linear verses hierarchical structure documents.
Recommended Books
1. Technical Report Writing, by Pauley and Riordan, Houghton Mifflin Company, 8th Edition. 2. Effective Technical Communication by Ashraf Rizvi, Tata McGraw-Hill.

5.2 Theory of Automata & Formal Languages

CSC-302 **Theory of Automata & Formal Languages**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Domain Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Explain and manipulate the different concepts in automata theory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc;	C2 (Understand)
CLO-2	Prove properties of languages, grammars and automata with rigorously formal mathematical methods	C3 (Apply)
CLO-3	Design of automata, RE and CFG	C6 (Create)
CLO-4	Transform between equivalent NFAs, DFAs and REs	C3 (Apply)

CLO-5	Define Turing machines performing simple tasks.	C3 (Apply)
CLO-6	Differentiate and manipulate formal descriptions of languages, automata and grammars with focus on regular and context-free languages, finite automata and regular expressions	C4 (Analyze)
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		
Recommended Books		
1. Introduction to computer theory, Daniel I. A. Cohen, 2nd Edition 2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011 3. An Introduction to Formal Languages and Automata, by Peter Linz, 4th edition, Jones & Bartlett Publishers, 2006 4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, 2005, New Age Publishers		

5.3 Numerical Analysis

CSC-352 **Numerical Analysis**	3(2-3) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Computer Science Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	The student would understand the fundamental concepts of Scientific Programming using programing Language(s)	C2 (Understand)
CLO-2	Use a computer algebra system to investigate and solve mathematical problems relating to integration, differential equations and approximation.	C3 (Apply)
Course Outline		

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

Sample Lab Experiments

The lab tasks focus on developing proficiency in Matlab for mathematical and computational problem-solving. The tasks begin with an introduction to Matlab, its commands, and installation. Students will then work with variables, arrays, and matrix declarations, and perform operations such as matrix and array concatenation. The course also covers complex numbers and workspace variables, alongside character strings and function calling. Students will learn to find the roots of polynomials and calculate derivatives. Additionally, they will use Matlab scripts to solve the Bisection method and the Newton-Raphson method. Tasks also include exploring Matlab functions related to matrices and arrays, and implementing solutions for Cramer's Rule and the Gaussian Elimination Method. Students will apply these techniques to solve simultaneous equations and explore advanced methods such as Gauss-Seidel and Successive Over-Relaxation (SOR). These tasks aim to provide a solid foundation in Matlab programming, enabling students to tackle various computational and mathematical challenges effectively.

Recommended Books

1. Numerical Analysis (9th Edition) by Richard L. Burden, J. Douglas Faires by Brooks/Cole Boston USA, 2011
2. Numerical Methods for Scientific Computing by J.H. Heinbockel Trafford Publishing USA, 2006
3. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116
4. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259
5. Prof. Sher Muhammad Ch. "Introduction to Statistical Theory-I, Ilmi Kitab Khana, Lahore.

5.4 Advance Programming

CSC-252 **Advance Programming**	3(2-3) **32 Theory + 32 labs**	**KNOWLEDGE AREA/ DOMAIN** Domain Elective
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Use the different elements of a visual programming language as building blocks to develop correct, coherent programs.	C3 (Apply)
CLO-2	Program using the fundamental software development process, including design, coding, documentation, testing, and debugging.	C3 (Apply)
CLO-3	Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs	C4 (Analyze)

Course Outline

Introduction to .NET, Persistence framework Layer, Object Relational Mapping,.NET framework details, JIT, Security, Memory Management, Garbage Collection, Mark, Compact Algorithm, Generations in garbage collection, CTS, CLS, Clavicula C# language basics constructs, Namespaces ,Enumerations ,Classes, Access modifiers, Properties ,Attributes, Methods, Loops, MechCollection's and MechCollection's (virtual and Abstract),Inheritance,VC#.NET controls 1,VC#.NET controls 2,Overview of MechCollection's to Data, Executing Commands, Working with Data, Choosing an ADO.NET Provider, Exception Handling, Practice of Layered Programming, Best Practices, Validation in Windows Practices, Validation between Connected & Disconnected Modes of Practices, Validation Procedures and Its Usage in Practices, Validation Procedures and Its Usage in ADO.NET,XML,XML manipulation in VC#.NET, Data Controls, The DataGrid View Control, The Report Viewer Control, Web Development (basic theory),Web Server / Container (basics),Web Request Cycle,Asp.net basics, HTML Controls, Server Controls, Difference between Html and Server Controls,Asp.net Controls, Security Implementation ,Session, Cookies, Validations, Controls, Security Controls in Controls, Security between:, grid View, List View, Repeater ,AJAX basics and asp.net ,Web Services in .NET,.NET Assemblies, Private Assemblies, Shared Assemblies, Multi-lingual Applications, Satellite Assembles ,.NET Remoting, Threading in .NET, Network Programming in .NET

Sample Lab Experiments

The lab tasks are structured to help students develop practical skills in C# and ASP.NET, with a focus on core programming concepts and web development. The first set of tasks includes programs such as finding the smallest single-digit factor, checking if a number is prime, and calculating the magnitude of a number. Next, students work on matrix operations, printing digits of an integer in words, and handling two-dimensional arrays of strings to identify the one with the most vowels. Additional tasks introduce structures, enumerations, constructors, inheritance, and polymorphism

in C#. Further tasks focus on partial classes, extension methods, and delegates in C#, as well as the use of common controls like Label, Text Box, Button, Combo Box, and List Box. Students will also create a Windows application for a registration form and another for calculating factorials and primes. In database-related tasks, students will use ADO.NET to establish database connections and work with Data Reader, Data Set, Data Adapter, and Data View objects. The course also covers static web page development using HTML and JavaScript validations, followed by ASP.NET controls such as Label, Button, and Textbox. Additional tasks involve cross-page and postback submissions, as well as dropdown lists and list boxes in ASP.NET. Students will also create a registration form with validation controls and develop a web service for arithmetic operations. Finally, tasks introduce assemblies, multithreading, and thread synchronization in C#. These tasks aim to provide a comprehensive understanding of both application and web development using C# and ASP.NET.
Recommended Books
1. Professional Visual C# by Wrox Series 2. Professional ASP.NET using C# by Wrox Series 3. Visual C#: How to Program, Deitel and Deitel, 6/e Edition, Prentice Hall / Pearson Education, 2017. 4. Programming in C# .NET, J.C. Bradley, A.C. Millspaugh, McGraw-Hill, 2014 5. Microsoft Visual C# 2013 Step by Step (Step by Step Developer), Sharp, J., 1st Edition (2013), Microsoft Press.

5.5 Financial Accounting

MGT-322 **Financial Accounting**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Elective Supporting
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Preparation of Financial Statements of Companies in accordance with statutory requirements of Companies Ordinance and International Financial Reporting Standards (IFRS) / International Accounting Standards (IAS) with appropriate notes to a preliminary extent,	C2 (Understand)
CLO-2	Compute working capital ratios for business sectors.	C3 (Apply)
CLO-3	Identify and explain Reasons for Profit Appropriation,	C4 (Analyze)
CLO-4	To learn about practical Implication of IAS-16, 18 and 38	C2 (Understand)
CLO-5	Application and selection of accounting techniques and procedures to specific circumstances like	C3 (Apply)

	leases, branches, departmental stores, consignment, joint venture and construction contracts.	
CLO-6	Prepare accounts and financial statements of joint venture, partnership, branches & departmental types of businesses,	C3 (Apply)
CLO-7	Identify and explain reasons why any loss/gain is debited or credited to retained earnings,	C4 (Analyze)
CLO-8	Prepare a statement of changes in Equity	C3 (Apply)
Course Outline		
Introduction to Accounting and its concepts. Recording Business Transactions: Journal, Ledger, Trial Balance. Preparation of Financial Statements: Balance Sheet, Income Statement, Completion of Accounting Cycle: Adjustments, Closing, Work Sheet Accounting for purchase and sales of merchandise. Receivable and payable, Inventories, Payroll Systems. Plant and Equipment: Acquisition, Depreciation, Disposal. Corporations: Organization and stock-holders equity, Operations, Earning per share and dividends		
Recommended Books		
1. International Accounting Standards (IAS) and International Financial Reporting Standards (IFRS) Latest Edition by ICAP 2. Kieso, Weygandt, and Warfield (Latest Edition) Intermediate Accounting, Latest Edition Wiley Higher Education. 3. Javed H. Zuberi , "Advanced Accounting", Latest Edition, Petiwala Book Depot. 4. An Insight into IFRSs by Mohyuddin Tahir		

6.1 Introduction to Marketing

MGT-351 **Introduction to Marketing**		3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Elective Supporting
After completion of this course students will be able to:			**Bloom's Taxonomy Level**
CLO-1	Understand core marketing concepts, including the marketing mix, consumer behavior, and market segmentation.		C2 (Understand)
CLO-2	Analyze market conditions and environmental factors to identify business opportunities and challenges.		C4 (Analyze)
CLO-3	Develop marketing strategies by applying segmentation, targeting, and positioning principles to achieve goals.		C5 (Evaluate)

CLO-4	Design and create promotional campaigns and marketing solutions incorporating modern trends and ethical practices.	C6 (Create)
Course Outline		
The Introduction to Marketing course provides a foundational understanding of key marketing concepts, including the marketing mix (4Ps: Product, Price, Place, Promotion), consumer behavior, and market segmentation, targeting, and positioning (STP). Students will explore the dynamics of the market environment, develop skills in creating marketing strategies, and learn about product lifecycle management, branding, and pricing strategies. The course also covers the role of distribution channels, supply chain management, and integrated marketing communication (IMC) techniques. Additionally, it addresses ethical considerations and emerging trends in marketing, such as digital marketing, sustainability, and the use of AI to adapt to the modern business landscape.		
Recommended Books		
1. Principles of Marketing (18th Edition) – Philip Kotler and Gary Armstrong 2. Marketing Management (16th Edition) – Philip Kotler and Kevin Lane Keller 3. Contemporary Marketing (19th Edition) – Louis E. Boone and David L. Kurtz 4. Essentials of Marketing (7th Edition) – Frances Brassington and Stephen Pettitt 5. Digital Marketing: Strategy, Implementation, and Practice (8th Edition) – Dave Chaffey and Fiona Ellis-Chadwick		

6.2 Compiler Construction

CSC-312 **Compiler Construction**	3(2-3) **32 Theory + 32 LAB**	**KNOWLEDGE AREA/ DOMAIN** Domain Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Understand the basic techniques used in compiler construction, such as lexical analysis, top-down and bottom-up parsing, context-sensitive analysis, and intermediate code generation.	C2 (Understand)
CLO-2	Understand the basic data structures used in compiler construction, including abstract syntax trees, symbol tables, three-address code, and stack machines.	C2 (Understand)

CLO-3	Design and implement a compiler using a software engineering approach.	C6 (Create)
CLO-4	Use tools like generators (e.g., Lex and Yacc) for compiler construction.	C3 (Apply)

Course Outline

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

Sample Lab Experiments

The lab tasks provide a comprehensive introduction to compiler construction, covering essential concepts and tools. The first task introduces the basic phases of a compiler, followed by a C program to scan and count the number of characters, words, and lines in a file. Students will then write C programs to implement NFAs (Non-deterministic Finite Automata) and DFAs (Deterministic Finite Automata) that recognize identifiers, constants, and operators of a mini-language. The next tasks involve implementing a lexical analyzer using tools like JLex or flex, as well as converting BNF (Backus-Naur Form) rules into Yacc form and writing code to generate a syntax tree. Students will also write a program to generate machine code from the abstract syntax tree produced by the parser. Additionally, there are tasks such as writing a Lex program to convert lowercase letters to uppercase (abc to ABC), and a Lex program to count the number of vowels and consonants in a given input string. Further tasks include the implementation of various parsers like the Predictive Parser, LR(0) Parser, SLR(1) Parser, LR(1) Parser, and LALR(1) Parser. The lab concludes with an introduction to YACC (Yet Another Compiler Compiler), a powerful tool for creating parsers. These tasks aim to give students hands-on experience with the core techniques used in building compilers and understanding their components.

Recommended Books

1. Compilers: Principles, Techniques, and Tools, A. V. Aho, R. Sethi and J. D. Ullman, Addison-Wesley, 2nd ed., 2006
2. Modern Compiler Design, D. Grune, H. E. Bal, C. J. H. Jacobs, K. G. Langendoen, John Wiley, 2003.
3. Modern Compiler Implementation in C, A. W. Appel, M. Ginsburg, Cambridge University Press, 2004.

6.3 Computer Architecture

CS-532 **Computer Architecture**	3(3-0) **32 Theory**	**KNOWLEDGE AREA/ DOMAIN** Domain Core

After completion of this course students will be able to:		Bloom's Taxonomy Level
CLO-1	Understand the function of major components of computer systems.	C2 (Understand)
CLO-2	Analyze the internal architecture and organization of the processor.	C4 (Analyze)
CLO-3	Validate the underlying theoretical concepts of computer architecture and organization through simulation.	C5 (Evaluate)

Course Outline

The course on Fundamentals of Computer Design covers essential concepts in computer organization and architecture, performance measurements, and the design of efficient computing systems. It begins by exploring the structure and function of computers, emphasizing performance considerations in system design. Students will study the evolution of the Intel x86 architecture and the ARM architecture, focusing on embedded systems. The course delves into Instruction Set Design principles, addressing modes, encoding, and a top-level view of computer function and interconnection. Key topics include computer components, bus interconnection, and main memory performance issues, as well as the impact of advanced DRAM organization and semiconductor memory on performance.

Further, the course investigates storage systems, including magnetic disk, RAID, and optical memory, with a focus on performance principles. The design and performance of cache memory are covered, alongside the Pentium 4 cache organization. I/O modules, including programmed I/O, interrupt-driven I/O, and DMA (Direct Memory Access), are examined, along with the external interface technologies like FireWire and InfiniBand.

The course also covers microprogrammed control, microinstruction sequencing, and exception handling techniques, including dynamic handling of exceptions to address performance issues. Students will learn about instruction sets, machine instruction characteristics, and the specific instruction sets of Intel x86 and ARM architectures, alongside processor structure, function, and instruction pipelining.

Key principles of Reduced Instruction Set Computers (RISCs) are explored, including the use of a large register file, compiler-based register optimization, and the concept of RISC pipelining. Topics such as instruction-level parallelism, superscalar processors, and design issues in Pentium and ARM Cortex-A8 are covered. Students will also explore parallel processing, symmetric multiprocessors, cache coherence, multithreading, chip multiprocessors, and clusters, as well as non-uniform memory access (NUMA) systems. The course also addresses performance issues in multicore computers, including hardware and software performance considerations and the organization of Intel x86 multicore systems.

This course aims to provide a deep understanding of computer architecture, performance optimization techniques, and the latest trends in modern processor design and multi-core systems.

Recommended Books
1. Computer Architecture: A Quantitative Approach by Hennessy & Patterson, Morgan & Kauffman Series (2006) Fourth Edition. 2. Stallings, "Computer Organization & Architecture", 8th Edition, Prentice HALL, 2008. 3. Computer Organization & Design: The Hardware/Software Interface By Patterson & Hennessy, Morgan & Kauffman Series (2008) 4thEdition.

6.4 Web Technologies

CSC-351 **Web Engineering**	3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Domain Elective
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Discuss how web standards impact software development.	C2 (Understand)
CLO-2	Describe the constraints that the web puts on developers.	C2 (Understand)
CLO-3	Design and implement a simple web application.	C6 (Create)
CLO-4	Review an existing web application against a current web standard.	C5 (Evaluate)

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.
Sample Lab Experiments

Introduction to compiler construction and phases of compiler, How to create Virtual Machines, Installation of Ubuntu, Introduction to Ubuntu, shell, software installation – exploring the OS, Introduction to UNIX, UNIX architecture, UNIX commands, Directory listing, UNIX File Management — Creating files, editing files, deleting files, copying files, counting words of file, displaying content, UNIX File Management— listing directory, changing file path, hidden files, UNIX Directory Management — Home directory, creating directory, changing directory, removing directories, UNIX Pipes and Filters – Grep command and parameters, UNIX Pipes and Filters – Sort command and parameters, UNIX Network Communication Utilities – Ping utility, FTP utility, UNIX Network Communication Utilities – Telnet utility, traceroute etc. Synchronization Problems – Producer Consumer Problem and its solution, Synchronization Problems – Dinner Philosopher's Problem, Synchronization Problems – Reader Writer Problem
Recommended Books
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.

7.1 Parallel & Distributed Computing

CSC-314 **Parallel & Distributed Computing**		3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Computer Science Core
After completion of this course students will be able to:			**Bloom's Taxonomy Level**
CLO-1	Learn about parallel and distributed computers.		C2 (Understand)
CLO-2	Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library		C3 (Apply)
CLO-3	Analytical modelling and performance of parallel programs.		C4 (Analyze)
CLO-4	Analyze complex problems with shared memory programming with openMP.		C4 (Analyze)
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)
Sample Lab Experiments
The course begins with an introduction to the Message Passing Interface (MPI), focusing on the basics and communication between MPI processes. Students explore advanced communication techniques within MPI, with an emphasis on collective operations such as Synchronization, Data Movement, and Collective Computation. The course also covers MPI Non-Blocking operations, providing students with the skills to efficiently manage communication without blocking processes. The second half of the course shifts to OpenMP (Open Multi-Processor API). Students familiarize themselves with OpenMP Directives and learn how to share work among threads using constructs such as the Loop Construct, Sections Construct, and Single Construct. The course also covers the use of Environment Variables within the OpenMP API to manage thread behaviors effectively. Additionally, students gain an understanding of Shared Memory Environments (SMP Programming) and Distributed Memory Environments (Cluster Programming), allowing them to apply parallel programming techniques across different system architectures.
Recommended Books
1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum and M. V. Steen, Prentice Hall, 2nd Edition, 2007 2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.

7.2 Mobile Application Development

CSC-353 Mobile Application Development	3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Domain Elective
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Discuss different architectures & framework for Mobile Application development.	1
CLO-2	Develop mobile applications using current software development environments.	3
CLO-3	Compare the different performance tradeoffs in mobile application development.	3

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 Email; 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..
Sample Lab Experiments
The course begins with an introduction to Java and explores both primitive and non-primitive data types. It covers Java operators and the manipulation of strings, along with conditional statements and loops. The role of adopter arrays in Android development is discussed, followed by methods and method overloading. The course integrates object-oriented concepts of Java with Android structures, and covers Java classes and objects, constructors, and destructors. Key object-oriented principles such as encapsulation, inheritance, and polymorphism are explored, alongside Java packages and their use in Android development, including creating and importing packages. Students learn about interfaces, multiple inheritance, wrapper classes, array lists, and exception handling in Java. Practical Android development is covered with button and text handlers, and data content manipulations. The course demonstrates the implementation of both primitive and non-primitive data types in Android applications, including using TextView and EditText classes for creating and managing objects. It introduces various views like TextView, Button, EditText, and ImageView, and the process of accessing images in Android. Additionally, the course focuses on adopter arrays, list and grid views, and the implementation of multiple activities in Android with data sharing between activities. Students also work on Android layouts, including Linear Layout, Relative Layout, and Grid Layout, and practice creating sophisticated user interfaces. The course further covers RecyclerView, CardView, and user-defined adapters, as well as working with dialog boxes, toasts, and fragments, including dynamic and static fragments and data passing between them. For practical application, the course includes building an Android food app, implementing shared preferences for login and logout functionality, and connecting to Firebase for data retrieval and storage. Students learn about animations and advanced UI design techniques. The final project integrates all concepts and techniques learned throughout the course.
Recommended Books
1. Professional Android application development, Reto Meier, Wrox Programmer to Programmer, 2015. 2. iOS Programming: The Big Nerd Ranch Guide, Conway, J., Hillegass, A., & Keur, C., 5 th Edition, 2014.

3. Android Programming: The Big Nerd Ranch Guides, Phillips, B. & Hardy, B., 2nd Edition, 2014.

8.1 Information Security

CS-553 Modern Programming Language	3(2-3) **32 Theory + 32 lab**	**KNOWLEDGE AREA/ DOMAIN** Computer Science Core
After completion of this course students will be able to:		**Bloom's Taxonomy Level**
CLO-1	Explain key concepts of information security such as design principles, cryptography, risk management, and ethics	2
CLO-2	Discuss legal, ethical, and professional issues in information security	2
CLO-3	Apply various security and risk management tools for achieving information security and privacy.	3
CLO-4	Identify appropriate techniques to tackle and solve problems in the discipline of information security.	4

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.

Sample Lab Experiments

Perform encryption, decryption using the following substitution techniques

i. Ceaser cipher
ii. Playfair cipher
iii. Hill Cipher
iv. Vigenere cipher.

Perform encryption and decryption using following transposition techniques

i. Rail fence
ii. Row & Column Transformation

Apply DES algorithm for practical applications. Apply AES algorithm for practical applications. Implement RSA Algorithm using HTML and JavaScript. Implement the Diffie-Hellman Key Exchange algorithm for a given problem. Calculate the message digest of a text using the SHA-1 algorithm. Implement the SIGNATURE SCHEME - Digital Signature Standard. Demonstrate

intrusion detection system (ids) using any tool eg. Snort or any other s/w. Automated Attack and Penetration Tools Exploring N-Stalker, a Vulnerability Assessment Tool.
Recommended Books
1. Computer Security: Principles and Practice, 3rd edition by William Stallings 2. Principles of Information Security, 6th edition by M. Whitman and H. Mattord 3. Computer Security, 3rd edition by Dieter Gollmann 4. Computer Security Fundamentals, 3rd edition by William Easttom 5. Official (ISC)2 Guide to the CISSP CBK, 3rd edition