MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING
(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

SCHEME & SYLLABUS
I/II SEMESTER MCA PROGRAM

2023 Scheme
(W.E.F 2023 Admission Students)

Vision

*"To attain perfection in providing **Globally Competitive Quality Education** to all our Students and also benefit the global community by using our strength in **Research and Development**"*

Mission

*"To establish world class educational institutions in their respective domains, which shall be **Centers of Excellence** in their stated and implied sense. To achieve this objective we dedicate ourselves to meet the challenges of becoming **Visionary and Realistic, Sensitive and Demanding, Innovative andPractical, Theoretical** and **Pragmatic**; ALL at the same time"*

INDEX

I/II SEMESTER		
Sl No	**Course Code**	**Course Title**
PROFESSIONAL CORE (PC)		
1	**23MCPC511**	Mathematical Foundation for Computer Applications
2	**23MCPC512**	Data Structures
3	**23MCPC513**	Database Management Systems
4	**23MCPC514**	Web Technologies
5	**23MCPC515**	Design and Analysis of Algorithms
6	**23MCPC516**	Software Engineering
7	**23MCPC521**	Operating System with LINUX Programming
8	**23MCPC522**	Object-Oriented Programming with Java
9	**23MCPC523**	Computer Networks
10	**23MCPC524**	Non-Relational Databases
PROFESSIONAL ELECTIVE (PE)		
11	**23MCPE5261**	Agile Software Development
12	**23MCPE5262**	Principles of Management and Organisational Behavior
13	**23MCPE5263**	Cryptography and Network Security
14	**23MCPE5264**	Computer Graphics with OpenGL
HUMANITIES (HM)		
15	**23MCHM525**	Research Methodology & IPR
SKILL ENHANCEMENT (SE)		
16	**23MCSE527**	Mini-Project
AUDIT COURSE (AU)		
17	**23MCAU517**	Basics of Computer Programming

I SEMESTER

SL No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)				Credits
					L	T	P	Hrs	CIE	SEE	Total	
1	23MCPC511	Mathematical Foundation for Computer Applications	PC	Mathematics	3	2	-	3	50	50	100	4
2	23MCPC512	Data Structures	PC	MCA	3	-	2	3	50	50	100	4
3	23MCPC513	Database Management Systems	PC	MCA	3	-	2	3	50	50	100	4
4	23MCPC514	Web Technologies	PC	MCA	4	-	-	3	50	50	100	4
5	23MCPC515	Design and Analysis of Algorithms	PC	MCA	4	-	-	3	50	50	100	4
6	23MCPC516	Software Engineering	PC	MCA	4	-	-	3	50	50	100	4
7	23MCAU517	Basics of Computer Programming *	AU	MCA	-	-	-	-	-	--	-	--
Total Credits												**24**

* Mandatory non-credit audit course

II SEMESTER

SL No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)				Credits
					L	T	P	Hrs	CIE	SEE	Total	
1	23MCPC521	Operating System with LINUX Programming	PC	MCA	3	-	2	3	50	50	100	4
2	23MCPC522	Object-Oriented Programming with Java	PC	MCA	3	-	2	3	50	50	100	4
3	23MCPC523	Computer Networks	PC	MCA	4	-	-	3	50	50	100	4
4	23MCPC524	Non-Relational Databases	PC	MCA	4	-	-	3	50	50	100	4
5	23MCHM525	Research Methodology & IPR	HM	MCA	4	-	-	3	50	50	100	4
6	23MCPE526X	Professional Elective 1	PE	MCA	3	-	-	3	50	50	100	3
7	23MCSE527	Mini-Project	SE	MCA	-	-	2	2.5	50	50	100	2
Total Credits												**25**

MATHEMATICAL FOUNDATION FOR COMPUTER APPLICATIONS			
Semester	I	CIE Marks	50
Course Code	23MCPC511	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:2:0	Exam Hrs	03
Total Hours	60 (40 hrs Theory+20 hrs Tutorials)	Credits	04

Course Learning Objectives:

This course is designed to

1. Lay a strong foundation of Sets, Relations and Functions to perform competent operations associated with them.
2. Impart knowledge of Mathematical logic empowering students to proficiently solve a variety of logical problems.
3. Introduce the basic principles of Graph theory and develop the ability to analyze graphs by exploring their properties.
4. Establish a foundation in statistical methods to effectively model and interpret data, make predictions, and draw meaningful conclusions.
5. Build a strong foundation in probability theory to solve problems involving random phenomena.

Module 1: Sets, Relations and Functions	No. of Hrs: 12

Sets, Relations and Functions: Basics of Set theory, Cartesian product of Sets, Relations and their properties, Relation matrix & Digraph of relations, Equivalence relations & Partitions.

Functions - Types of functions, Function composition and Inverse function, Applications of sets, relations and functions to solve simple real life problems. Introduction to mathematical computation using MATLAB.

Self-study: Representing sets, performing set operations, and investigating Relation matrix using MATLAB.

Module 2: Mathematical logic	No. of Hrs: 12

Mathematical Logic: Basic connectivity and Truth table, Logical equivalences, Quantifiers, Predicative Logic, Free and Bound variables, Rules of inference, Proofs of theorems - Induction, Direct, Indirect, and Proof by Contradiction.

Self-study: Performing logical operations using MATLAB.

Module 3: Graph theory	No. of Hrs: 12

Graph theory: Graphs & Graph models, Subgraphs, Complement and Graph Isomorphism, Connectivity & Shortest Path algorithms - Depth First Search, Breadth First Search and problems, Euler's & Hamiltonian paths, Graph theory applications in solving real-life problems.

Self-study: Representing graphs, and determining its combinatorial properties using MATLAB

Module 4: Statistics	No. of Hrs: 12

Statistics: Curve fitting by the method of least squares, fitting of curves – Polynomial and Exponential. Correlation and Linear regression, Applications of Curve fitting and Linear Regression in addressing real-life problems.

Self-study: Computing correlation coefficient, fitting of curves using MATLAB.

Module 5: Probability	No. of Hrs: 12

Probability: Basic concepts of probability, properties of probability, Conditional probability, Bayes' theorem, Application of probability in solving real-life problems.

Self-study: Computing conditional probability, expectation and variance using MATLAB

Course Outcomes:

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

CO1: Illustrate the concepts of Sets, Relations & Functions, Mathematical logic, Graph theory, Statistics and Probability.

CO2: Apply the concepts of Sets, Relations & Functions, Mathematical logic, Graph theory, Statistics and Probability to solve related problems.

CO3: Solve real-life problems based on the concepts of Sets, Relations & Functions, Mathematical logic, Graph theory, Statistics and Probability.

CO4: Make use of MATLAB to perform mathematical computations related to sets, relations and functions, Mathematical logic, curve fitting, linear regression, graphs in MATLAB.

TEXT BOOKS

1. Kenneth H Rosen, *Discrete Mathematics and its Applications*, (8th ed), Tata McGraw-Hill Education Private Limited, 2023.
2. Ronald E. Walpole, Sharon L Myers, *Probability and Statistics for Engineers and Scientists*, (9th ed), Pearson Education, 2022.

REFERENCE BOOKS

1. Sheldon Ross, *A First Course in Probability*, (10th ed), Pearson, 2023.
2. J.K Sharma, *Discrete Mathematics*, (4rd ed), Macmillan Publishers India, 2018.
3. Oliver C. Ibe, *Fundamentals of Applied Probability and Random Process*, (2nd ed), Elsevier Academic Press, 2023.

WEB LINKS:

1. https://nptel.ac.in/courses/111107058
2. https://onlinecourses.nptel.ac.in/noc20_cs82/preview
3. https://archive.nptel.ac.in/courses/111/105/111105042/

DATA STRUCTURES			
Semester	I	CIE Marks	50
Course Code	23MCPC512	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:2	Exam Hrs	03
Total Hours	64 (40 hrs Theory+24 hrs Lab)	Credits	04

Course Learning Objectives:

This course is designed to

1. Learn the fundamentals of data structures.
2. Provide the knowledge of basic data structures and their implementations.
3. Familiarize students in writing efficient programs using appropriate data structures.
4. Develop skills to apply the knowledge of data structures in problem solving.

Module 1 : Classification of Data Structures	No. of Hrs: 13

Primitive and Non-Primitive, Linear and Nonlinear; Data structure Operations, Stack: Definition, Representation, Operations and Applications: Polish and reverse polish expressions, Infix to postfix conversion, evaluation of postfix expression, infix to prefix, postfix to infix conversion. Recursion: Factorial, GCD, Fibonacci Sequence, Tower of Hanoi.

Laboratory Component

1. Write a C program to implement stack with the following operations:
   i. Push an element on to stack.
   ii. Pop an element from the stack.
2. Implement a program in C for converting a given Infix Expression to Postfix Expression.
3. Write a C program to find the factorial of a given number using recursion.

Module 2: Queue and its Applications	No. of Hrs: 12

Queue: Definition, Representation, Queue Variants: Simple Queue, Circular Queue, Priority Queue, Double-Ended Queue; Applications of Queues, Operations on Queue, Programming Examples.

Laboratory Component

2. Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Perform the following operations:
   i. Insert
   ii. Delete
   iii. Display.

Module 3: Linked List	No. of Hrs: 13

Limitations of array implementation, Memory Management: Static (Stack) and Dynamic (Heap) Memory Allocation, Memory management functions. Definition, Representation, Operations: getnode() and freenode() operations, Types: Singly Linked List. Linked list as a Data Structure, Inserting and removing nodes from a list, Linked implementations of stacks and queues, Header nodes.

Laboratory Component

1. Write a C program to simulate the working of a singly linked list with the following operations:
   a. Insert
   b. Delete
   c. Display
2. Develop a C program to demonstrate the operations on a stack using singly linked lists.

Module 4 : Trees and Graph	No. of Hrs: 13

Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals – In-order, Post-order, Pre-order

Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph, Operations, Traversal methods: Breadth First Search and Depth First Search.

Laboratory Component

1. Develop C programs on binary trees. (Construct a binary search tree and traverse the tree using all the methods i.e., In-order, Post-order, Pre-order).

Module 5 : Sorting and Searching	No. of Hrs: 13

Brute Force: Selection Sort and Bubble Sort, Sequential Search, Divide-and-Conquer: Merge sort, Quicksort, Binary Search, Decrease-and-Conquer: Insertion Sort, Shell sort, Sequential search, Indexed sequential search, Binary search, Binary Tree Search.

Laboratory Component

1. Write a C program to implement the following search techniques:
   i. Linear Search
   ii. Binary Search
2. Write a C program to implement the following sorting algorithms using user-defined functions:
   i. Bubble sort (Ascending order)
   ii. Selection sort (Descending order)

Course Outcomes:

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

CO1: Identify the basic concepts of data structures, its applications and dynamic memory management.

CO2: Compare the different sorting and searching techniques.

CO3: Illustrate on the various operations with trees, graphs and traversal mechanisms.

CO4: Examine the operational aspects of stacks, queues and linked lists in problem solving.

TEXT BOOKS

1. Ellis Horowitz and Sartaj Sahni, *Fundamentals of Data Structures in C,* (2nd ed), Universities Press, 2014.
2. Seymour Lipschutz, *Data Structures Schaum's Outlines,* (Revised 1st ed), McGraw Hill, 2014.
3. Bala Guruswamy, *Programming in ANSI C*, (8th ed), McGraw Hill, 2019.

REFERENCE BOOKS

1. Gilberg & Forouzan, *Data Structures: A Pseudocode approach with C*, (2nd ed), Cengage Learning, 2014.
2. Reema Thareja, *Data Structures using C*, (3rd ed), Oxford Press, 2012.
3. Kenneth A Berman and Jerome L Paul, *Algorithms*,(2nd ed), Cengage Learning India Pvt Ltd, 2002.

WEB LINKS:

1. https://nptel.ac.in/courses/106/102/106102064/
2. www.javatpoint.com/tree-vs-graph-data-structure
3. www.mygreatlearning.com/blog/data-structures-using-c/#binary-tree
4. https://archive.nptel.ac.in/courses/106/102/106102064/

DATABASE MANAGEMENT SYSTEMS			
Semester	I	CIE Marks	50
Course Code	23MCPC513	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:2	Exam Hrs	03
Total Hours	64 (40 hrs Theory+24 hrs Lab)	Credits	04

Course Learning Objectives:

This course is designed to

1. Provide a strong foundation in database concepts, technology, and practice.
2. Practice SQL programming through a variety of database problems.
3. Demonstrate the use of concurrency and transactions in a database system.
4. Build database applications for real-world problems.

Module 1 : Introduction to Databases	No. of Hrs: 12

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications. Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. Conceptual Data Modelling using Entities and Relationships: Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams.

Laboratory Component

1. Consider the following schema: STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA).
   Execute the following queries:
   a. Update the column total by adding the columns mark1, mark2, mark3.
   b. Find the GPA score of all the students.
   c. Find the students who were born on a particular year of birth from the date_of_birth column.
   d. List the students who are studying in a particular branch of study.
   e. Find the maximum GPA score of the student branch-wise.
   f. Find the students whose name starts with the alphabet “S”.
   g. Find the students whose name ends with the alphabet “AR”.
   h. Delete the student details whose USN is given as 1001.
2. Consider the following database of student enrollment in courses and books adopted for each course.
   STUDENT (regno#: string, name: string, major: string, bdate: date)
   COURSE (course#: int, cname: string, dept: String)
   TEXT(book_ISBN#: int, book_title: string, publisher: string,author:string)
   ENROLL (regno#: string, course#: int, sem: int, marks: int)
   BOOK_ADOPTION (course#: int, sem: int, book_ ISBN: int)
   Execute SQL queries for the following :
   a. List out the student details, and their course details. The records should be ordered in a

semester wise manner.

b. List out the student details under a particular department whose name is ordered in a semester wise.
c. List out all the book details under a particular course.
d. Find out the Courses in which the number of students studying will be more than 2.
e. Find out the Publisher who has published more than 2 books.

Module 2: Relational Model	No. of Hrs: 13

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra. Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping.

Laboratory Component

1. Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries. Consider a Cricket Tournament "ABC CUP" organized by an organization. In the tournament there are many teams contesting each having a Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers,age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode). A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player. Execute the following queries:
   a. Display the youngest player (in terms of age) Name, Team name, age in which he belongs to the tournament.
   b. List the details of the stadium where the maximum number of matches were played.
   c. List the details of the player who is not a captain but got the man_of _match award at least in two matches.
   d. Display the Team details who won the maximum matches.
   e. Display the team name where all its won matches played in the same stadium.

Module 3: Structured Query Language	No. of Hrs: 13

SQL: SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL. Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures.

Laboratory Component

1. A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state, Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno, city, state, pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidate is uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party. There are many parties. Each party is uniquely identified by using Party_id, having Party_Name, Party_symbol. A candidate can contest from many constituencies under the same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belonging to different parties. Each voter votes only one candidate of his/her constituency.

   Execute the following queries:

   a. List the details of the candidates who are contesting from more than one constituency which belong to different states.
   b. Display the state name having the maximum number of constituencies.
   c. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If the voter's age is at least 18 years old, then insert the tuple into the voter else display the “Not an eligible voter msg”.
   d. Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure.
   e. Create a TRIGGER to UPDATE the count of “ Number_of_voters” of the respective constituency in the “CONSTITUENCY” table , AFTER inserting a tuple into the “VOTERS” table.

Module 4 : Normalization	No. of Hrs: 13

Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form. Examples of normal forms. Normalization Algorithms: Inference Rules, Nulls Dangling tuples and alternate Relational Designs dependencies and 4NF.

Laboratory Component

1. Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places. Each Tourist place is identified by using tourist_place_id, having a name, belonging to a state, capital city of that state, history. There are many Tourists visiting tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple email ids. A tourist visits many Tourist places, it is also required to record the visited_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either on the same date or at different dates. Queries:
   a. List the state name which has the maximum number of tourist places.
   b. List details of Tourist places where the maximum number of tourists visited.
   c. List the details of tourists visiting all tourist places from the state "KARNATAKA".
   d. Display the details of the tourists who visited at least one tourist place of the state, but visited all tourist places in all states.
   e. Display the details of the tourist place visited by the tourists of all countries.

Module 5: Transaction Processing and Concurrency Control	No. of Hrs: 13

Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, characterizing schedules based on recoverability, characterizing schedules based on Serializability, Transaction support in SQL.

Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multi version Concurrency control techniques, Validation Concurrency control techniques.

Course Outcomes:

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

CO1: Identify and define database objects, enforce integrity constraints on a database using RDBMS.

CO2: Recognize SQL queries for database manipulation and summarize the basics of query evaluation.

CO3: Use simple database systems to relate the concept of transaction, concurrency control and recovery.

CO4: Investigate applications to interact with databases, tuples and domain relational expressions from queries.

TEXT BOOKS

1. Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems.* (7th ed), Pearson, 2017.
2. Ramakrishnan and Gehrke, *Database Management Systems,* (3rd ed), McGraw Hill, 2014.

REFERENCE BOOKS

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, *Database System Concepts.* (6th ed), Tata Mcgraw Hill Education Private Limited, 2011.

WEB LINKS:

1. https://www.youtube.com/watch?v=3EJlovevfcA
2. https://www.youtube.com/watch?v=9TwMRs3qTcU
3. https://www.youtube.com/watch?v=ZWl0Xow304I
4. https://www.youtube.com/watch?v=4YilEjkNPrQ
5. https://www.youtube.com/watch?v=CZTkgMoqVss

WEB TECHNOLOGIES			
Semester	I	CIE Marks	50
Course Code	23MCPC514	SEE Marks	50
Teaching Hours/Week (L:T:P)	(4:0:0)	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. Learn some basic tags of XHTML5.
2. Implement simple web pages using XHTML5 and CSS.
3. Develop dynamic documents using JavaScript with CSS.
4. Create dynamic web pages using AngularJS, ReactJS code and connection to a server.

Module 1 : Introduction to XHTML5 and CSS	No. of Hrs: 10

Web browsers, web servers, MIME, URL, HTTP Introduction to XHTML5 tags, Basic syntax and structure, text markups, images, lists, tables, progress, Media tags-audio and video, forms, frames. Introduction to CSS Levels of CSS, Selectors, Font, color and Text Properties, BOX Model, Span and Div tags.

Module 2: JavaScript	No. of Hrs: 10

Introduction to JavaScript, Control statements, Arrays and functions, Pattern matching, Element Access, Event Handling.

Module 3: Angular JS	No. of Hrs: 10

Introduction to AngularJS, Directives, Expressions, Directives, Controllers, Filters, Services, Events, Forms, Validations, Examples.

Module 4 : Introduction to the MERN stack	No. of Hrs: 10

Introduction, The MVC Architectural Pattern, MERN Components React, Node.js, Express, MongoDB, Advantages of MERN, Isomorphic.

Module 5 : Understanding React and Web Server	No. of Hrs: 12

Welcome to React- Obstacles and Roadblocks, Reacts Future, Server setup, NVM, Node Js, Project, NPM, Express, Build time JSX compilation- Separate Script File, Transform, Automate, React Library, React Components- React classes, Composing components, passing data using properties, property validation, using children's Dynamic composition.

Course Outcomes:

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

CO1: Describe the concepts of CSS and XHTML5.

CO2: Explain the process of creating a web page using XHTML5, JavaScript and CSS.

CO3: Produce dynamic web pages using AngularJS.

CO4: Make use of the dynamic connectivity between ReactJS code and a Server.

TEXT BOOKS

1. Chris Bates, *Web Programming*, (3rd ed), Wiley Publications, 2007.
2. Robert W. Sebesta, *Programming the World Wide Web*, (4th ed), Pearson education, 2012.
3. *HTML5 Black Book*, (3rd ed), Dreamtech Press, 2019.
4. Anthony Accomazzo, Ari Lerner, Nate Murray, Clay Allsopp, David Gutman, and Tyler McGinnis, *Fullstack React: The Complete Guide to ReactJS and Friends*, (1st ed), Fullstack.io, 2017.
5. Greg Lim, *Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App*, (1st ed), Amazon Digital Services LLC, 2021.

REFERENCE BOOKS

1. Uttam K Roy, *Web Technologies*, (1st ed), Oxford University Press, 2010.
2. M. Deitel, P.J. Deitel, A. B. Goldberg, *Internet & World Wide Web: How to Program*, (5th ed), Pearson Education, 2008.

WEB LINKS:

1. www.w3schools.com/JQuery/default.asp
2. https://www.coursera.org/specializations/web-applications
3. https://legacy.reactjs.org/docs/getting-started.html
4. www.mongodb.com/languages/mern-stack-tutorial

DESIGN AND ANALYSIS OF ALGORITHMS			
Semester	I	CIE Marks	50
Course Code	23MCPC515	SEE Marks	50
Teaching Hours/Week (L:T:P)	(4:0:0)	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. Acquire knowledge of basic algorithms and their efficiency analysis.
2. Analyze the asymptotic performance of algorithms.
3. Introduce different algorithm design paradigms with illustrative problems.
4. Synthesize efficient algorithms in common engineering design situations.

Module 1 : Introduction	No. of Hrs: 12

Introduction: Notion of an Algorithm, Algorithm Specification, Analysis Framework, Performance Analysis: Space complexity, Time complexity and notation (o), Mathematical analysis of Non-Recursive and recursive Algorithms), Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Little-oh) with Examples, Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. Fundamental Data Structures: Stacks, Queues, Graphs, Trees, Sets and Dictionaries.

Module 2: Divide and Conquer	No. of Hrs: 10

Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum, Merge sort, Quick sort, Strassen's matrix multiplication, Advantages and Disadvantages of divide and conquer. Decrease and Conquer Approach: Topological Sort. Transform and Conquer Approach: Heaps and Heap Sort.

Module 3: Algorithms	No. of Hrs: 10

Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines. Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm. Single source shortest paths: Dijkstra's Algorithm. Optimal Tree problem: Huffman Trees and Codes.

Module 4 : Dynamic Programming	No. of Hrs: 10

Dynamic Programming: General method with Examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem, Bellman-Ford Algorithm, Travelling SalesPerson problem

Module 5: Backtracking	No. of Hrs: 10

Backtracking: General method, N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. Programme and Bound: Assignment Problem, Travelling SalesPerson problem, 0/1 Knapsack problem: LC Programme and Bound solution, Probabilistic and Randomized Algorithms: Probabilistic Algorithms Randomizing deterministic Algorithms: MonteCarlo Algorithm, Biased Monte Carlo Algorithms: A Montecarlo algorithm for testing polynomial quality.

Course Outcomes:

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

CO1: Describe the approaches used in estimating the time and space complexity of algorithms.

CO2: Discuss Brute Force, Divide and Conquer algorithms and measure their performance.

CO3: Classify the different Decrease and Conquer algorithms and discuss the space and time trade-offs techniques.

CO4: Characterize the features of various graphical problems with the help of a suitable algorithmic technique.

CO5: Evaluate the limitations of Backtracking, Branch & Bound technique in solving problems.

TEXT BOOKS

1. Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, (2nd ed), Pearson, 2009.
2. Ellis Horowitz, Sartaj Sahni and Rajasekaran, *Computer Algorithms/C++*, (2nd ed), Universities Press, 2014.
3. Kenneth A Berman and Jerome L Paul, *Algorithms :Foundations and Design Strategies*, (1st ed), Algorithms, 2017.

REFERENCE BOOKS

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*, (3rd ed), MIT Press, 2009.
2. S. Sridhar, *Design and Analysis of Algorithms*, (2nd ed), Oxford (Higher Education), 2023.

WEB LINKS:

1. lms.vtu.ac.in/econtent/courses/CSE/06CS43/index.php
2. https://nptel.ac.in/courses/106/101/106101060/
3. http://cse01-iiith.vlabs.ac.in/

SOFTWARE ENGINEERING			
Semester	I	CIE Marks	50
Course Code	23MCPC516	SEE Marks	50
Teaching Hours/Week (L:T:P)	4:0:0	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. Outline software engineering principles and activities involved in building large software programs.
2. Explain the fundamentals of Object-Oriented concepts.
3. Identify ethical and professional issues and explain why they are of concern to software engineers.
4. Describe the process of requirements gathering and requirements classification.
5. Build effective software engineering applications in real-time.

Module 1: Introduction	No. of Hrs: 10

Professional software development: Software engineering ethics. Software processes: Software process models, Process activities, Coping with change, The rational unified process.

Module 2: Requirements	No. of Hrs: 12

Requirements engineering: Functional and non-functional requirements, The Software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management.

System modeling: Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering.

Module 3: Design	No. of Hrs: 10

Architectural design: Architectural design decisions, Architectural views, Architectural patterns, Application architecture.

Software testing: Development testing, Test-driven development, Release testing, User testing.

Module 4 : Distributed Software Engineering	No. of Hrs: 10

Distributed software engineering: Distributed systems issues, Client–server computing, Architectural patterns for distributed systems, Software as a service.

Service-oriented architecture: Services as reusable components, Service engineering, Software development with services.

Module 5: Project Management	No. of Hrs: 10

Project management: Risk management, Managing people, Teamwork.

Project planning: Software pricing, Plan-driven development, Project scheduling, Estimation techniques. Quality management: Software quality, Software standards, Reviews and inspections, Software measurement and metrics.

Course Outcomes:

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

CO1: Recognize the importance of the various software models while developing effective applications.
CO2: Explain the need for an SRS while designing efficient software applications.
CO3: Illustrate the principles of software testing and classify their types while testing applications.

CO4: Analyze the scope of a project plan to maintain a good project management system.

TEXT BOOKS:

1. Ian Sommerville, *Software Engineering*, (9th ed), Pearson Education, 2012.
2. Pankaj Jalote, *An Integrated Approach to Software Engineering*, (3rd ed), Springer New York, 2005.

REFERENCE BOOKS:

1. Roger S. Pressman, Bruce R Maxim, *Software Engineering-A Practitioner's approach*, (9th ed), Tata McGraw Hill, 2023.
2. Stephan R. Schach, *Object Oriented Software Engineering*, (1st ed), Tata McGraw Hill, 2008.
3. Michael Blaha, James Rumbaugh, *Object Oriented Modelling and Design with UML*, (2nd ed), Pearson Education, 2005.

WEB LINKS:

1. http://en.wikipedia.org/wiki/Software_engineering
2. http://www.cmcrossroads.com/bradapp/links/swe-links.html

BASICS OF COMPUTER PROGRAMMING			
Semester	I	CIE Marks	-
Course Code	23MCAU517	SEE Marks	-
Teaching Hours/Week (L:T:P)	-	Exam Hrs	-
Total Hours	35 (25 hrs Theory + 10 hrs Lab)	Credits	-

Course Learning Objectives:

This course is designed to

1. Learn the basic components of a computer system and their characteristics.
2. Know the traditional programming model and to write programs with the C language.
3. Identify the elements of modern instruction sets and their impact on processor design.

Module 1 : Basics of C Programming	No. of Hrs: 05

C Programming: Decision making, control structures and arrays C Structure, Data Types, Input-Output Statements, Decision making with if statement, simple if statement, the if..else statement, nesting of if..else statements, the else.if ladder, the switch statement, the ?: operator, the goto statement, the break statement, The while statement, the do...while statement, the for statement, nested loops, jumps in loops, the continue statement.

Arrays: One dimensional and two dimensional arrays, declaration and initialization of arrays, reading, writing and manipulation of above types of arrays, Programming Examples.

Laboratory Component

1. Program to check whether the given character is Lowercase or Uppercase or a Special Character.
2. Program to swap two numbers without using a third variable
3. Write functions to implement string operations such as compare, concatenate, string length. Convince the parameter passing techniques.

Module 2: Structures	No. of Hrs: 05

Structures: Defining a structure, declaring structure variables, accessing structure members, structure initialization, copying and comparing structure variables, operations on individual members, array of structures, structures within structures, structures and functions, Unions, size of structures, Programming Examples.

Laboratory Component

1. Implement structures to read, write and compute average marks and the students scoring above and below the average marks for a class of N students.
2. Program to store data in structures dynamically

Module 3: Pointers	No. of Hrs: 05

Pointers in C: Declaring and accessing pointers in C, Pointer arithmetic, Functions, Call by value, Call by reference, Pointer as function arguments, recursion, Passing arrays to functions, passing strings to functions, Functions returning pointers, Pointers to functions, Programming Examples.

Laboratory Component

1. Develop a program using pointers to compute the sum,mean and standard deviation of all elements stored in an array of N real numbers.

2. Write a program to find the reverse of a string using pointers.

Module 4 : Binary System and Combinational Logic	No. of Hrs: 05

Binary Systems and Combinational Logic: Digital Computers and Digital Systems, Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, subtraction using r's and r-1 complements, Binary Code, Binary Storage and Registers, Binary Logic, Integrated Circuits, Digital Logic Gates, Programming Examples.

Module 5: Basic Structure of a Computer	No. of Hrs: 05

Basic Structure of Computer Hardware and Software Computer Types, Functional Units, Basic Operational Concepts, Bus structure, Software, Performance, Multiprocessing and Multi computers, Machine Instruction: Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Interrupts.

Course Outcomes:

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

CO1: Define the key concepts introduced in C programming by writing and executing programs.

CO2: Compare the concepts of structures and pointers for the given application/problem.

CO3: Use the concepts of single/multi-dimensional arrays for a given problem.

CO4: Analyze how memory organization, operations, instruction sequencing and interrupts are useful in executing a given program.

TEXT BOOKS

1. Balaguruswamy, *Programming in ANSI C*, (8th ed), McGraw Hill Education, 2019.
2. Herbert Schild, *The C Complete Reference* (4th ed), McGraw Hill Education, 2000.
3. Yashwant Kanetkar, *Let us C*, (19th ed), BPB Publications, 2022.

REFERENCE BOOKS

1. M. Morris Mano, *Digital Logic and Computer Design*, (1st ed), Pearson, 2012
2. Carl Hamacher, Zvonko Vranesic Safwat Zaky, (2012), *Computer Organization*, (5th ed), Tata McGraw-Hill, 2012.

WEB LINKS:

1. https://www.javatpoint.com/c-programming-language-tutorial
2. https://www.coursera.org/courses?query=c%20programming
3. https://www.udemy.com/topic/c-programming/
4. https://www.cuemath.com/numbers/binary-number-system/

OPERATING SYSTEM WITH LINUX PROGRAMMING			
Semester	II	CIE Marks	50
Course Code	23MCPC521	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:2	Exam Hrs	03
Total Hours	64 (40 hrs Theory+24 hrs Lab)	Credits	04

Course Learning Objectives:

This course is designed to

1. Learn the basic concepts and structure of operating systems.
2. Understand the different process scheduling techniques.
3. Analyze the basics and management of file systems.
4. Know the basics of shell programming.

Module 1 : Introduction	No. of Hrs: 12

Overview: Introduction to Operating Systems, Computer System Architecture, Operating System Structure, Operating System Operations, Distributed Systems, Special purpose systems, Computing environments, Open-source operating system.

System Structures: Operating System Services, System Calls, Types of System Calls, System Programs, Operating system structure, Virtual Machines, Operating system Generation, System boot.

Laboratory Component

1. Execution of LINUX commands.

Module 2: Process Management	No. of Hrs: 13

Process Management: Process concept, process state, process control block, Process Scheduling Process Scheduling: Basic concepts, Scheduling criteria, Scheduling Algorithms: FCFS, SJFS, Priority scheduling, Round Robin Scheduling, Multi-level queue scheduling, Multilevel feedback queue scheduling.

Multithreaded Programming: Overview, Multithreaded Models, Threading Issues, Operating-system Examples.

Laboratory Component

1. Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.
   a. FCFS
   b. SJF
   c. Round Robin (pre-emptive)
   d. Priority.

Module 3: Synchronization and Deadlocks	No. of Hrs: 13

Process Synchronization: Critical section problem, Peterson's Solution, Synchronization hardware, Semaphore, classic problems of synchronization, Monitors, Synchronization Examples

Deadlocks: System model, Deadlock Characterization, Methods for handling deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from deadlock.

Laboratory Component

1. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.

2. Write a C program to simulate the concept of Dining-Philosophers problem.
3. Write a C program to simulate producer-consumer problem using semaphores.

Module 4 : The File System	No. of Hrs: 13

The File System: The File, What's in a File name? The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, The Unix File System. Basic File Attributes: ls options, File Ownership, File Permissions, chmod, Directory Permissions, Changing the File Ownership More File Attributes: File Systems and Inodes, Hard Links, Symbolic Links, The Directory, umask, Modification and Access Times, find. The Shell: The Shell's Interpretive Cycle, Shell Offerings, Pattern Matching-The Wild-cards, Escaping and Quoting, Redirection: The Three Standard Files, Two Special Files: /dev/null and /dev/tty, pipes, tee: Creating a Tee, Command Substitution.

Laboratory Component

1. Write a C program to simulate the following file organization techniques
   a. Single level directory
   b. Two level directory
   c. Hierarchical
2. Execution of file system commands.

Module 5: Shell Programming	No. of Hrs: 13

Essential Shell Programming: Shell Variables, Environment Variables, Shell Scripts, read, Using Command Line Arguments, exit and exit status of command, The Logical Operators, The if Conditional, using test and [] to Evaluate Expression, The case Conditional, expr, while: looping, for: looping with a list, set and shift.

Laboratory Component

1. Implementation of Shell programs using if condition, case and loops.

Course Outcomes:

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

CO1: Describe the basic structure of an Operating System and the concepts involved in process management.

CO2: Compare the performance of different scheduling algorithms along with the policies for concurrency and deadlock management.

CO3: Categorize the system calls used for process management and file management.

CO4: Differentiate between the LINUX commands for memory management, file management and directory management.

TEXT BOOKS

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating Systems Principles*, (8th ed), Wiley India, 2009.
2. Sumitabha Das, *UNIX Concepts and Applications*, (4th ed), Tata McGraw Hill, 2006.

REFERENCE BOOKS

1. D. M. Dhamdhere, *Operating Systems – A Concept Based Approach*, (2nd ed), Tata McGraw – Hill, 2006.

2. P. C. P. Bhatt, *Operating Systems*, (2nd ed), PHI, 2006.
3. W. Richard Stevens Stephen A. Rago, *Advanced Programming in the UNIX Environment*, (3rd ed), Addison Wesley, 2013.
4. Harvey M Deital, *Operating Systems*, (3rd ed), Addison Wesley, 1990.

WEB LINKS:

1. https://www.coursera.org/learn/akamai-operating-systems
2. https://onlinecourses.nptel.ac.in/noc20_cs04/preview
3. https://www.udemy.com/course/the-complete-operating-systems-course-from-zero-to-expert/
4. https://www.javatpoint.com/operating-system

OBJECT-ORIENTED PROGRAMMING WITH JAVA			
Semester	II	CIE Marks	50
Course Code	23MCPC522	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:2	Exam Hrs	03
Total Hours	64 (40 hrs Theory+24 hrs Lab)	Credits	04

Course Learning Objectives:

This course is designed to

1. Make students familiar with the basic object-oriented programming concepts and apply them in problem solving.
2. Use object-oriented programming concepts to solve real-world problems.
3. Explain the concept of class and objects with access control to represent real world entities.
4. Illustrate the behavior of programs involved in basic programming constructs like control structures, overloading, overriding, constructors, string handling and garbage collection.
5. Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.

Module 1 : OOPS Concepts and Java Programming	No. of Hrs: 13

OOP Concepts: Classes and objects, data abstraction, encapsulation, inheritance, polymorphism, procedural and object oriented programming paradigm. Java programming: History of java, comments data types, variables, constants, scope and lifetime of variables, operators, operator hierarchy, expressions, type conversion and casting, control flow statements, jump statements, simple java stand-alone programs, arrays, console input and output, formatting output, constructors methods, static fields and methods, access control, overloading methods and constructors, recursion.

Laboratory Component

1. Write a Java program to print the following triangle of numbers.

   1

   1 2

   1 2 3

   1 2 3 4

   1 2 3 4 5
2. Write a Java program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use a while loop. (Hint Fact of 4 = 4*3*2*1)

Module 2: Multiple inheritance and interface	No. of Hrs:13

Inheritance: Inheritance hierarchies, super and subclasses, member access rules, super keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: dynamic binding, method overriding, abstract classes and methods, defining an interface, implement interfaces, accessing implementations through interface references, extending interface.

Laboratory Components

1. Write a Java program:
   a. To find the area and circumference of the circle by accepting the radius from the user.
   b. To accept a number and find whether the number is Prime or not.
2. Write a Java program to demonstrate Multiple inheritance using interfaces and to calculate the area of a rectangle and triangle.

Module 3: Exception Handling	No. of Hrs:12

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception subclasses.

Laboratory Components

1. Write a Java program to demonstrate Constructor Overloading and Method Overloading.
2. Write a Java program to handle divide by zero Exception.

Module 4 : Multi-Threaded programming	No. of Hrs:13

Multithreading fundamentals: The Thread Class and Runnable Interface, Creating Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), suspending, Resuming and stopping Threads.

Laboratory Components

1. Write a Java program to create multiple threads using different thread methods.

Module 5: GUI Programming And Applets	No. of Hrs:13

GUI Programming with Java: The AWT class hierarchy, introduction to swing, hierarchy for swing components.

Overview of some swing components: JButton, JLabel, JTextField, JTextArea, simple applications. **Layout management**: Layout manager types, border, grid and flow.

Applets: Inheritance hierarchy for applets, life cycle of an applet, passing parameters to applets.

Laboratory Components

1. Write a Java applet program which handles keyboard events.

Course Outcomes:

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

CO1: Recall the various object-oriented programming concepts and their importance while writing Java programs.

CO2: Summarize the different types of inheritance with its need and usage.

CO3: Apply Exception handling concepts to write effective programs in Java.

CO4: Distinguish between AWT and Swing components while creating Graphical User Interfaces.

TEXT BOOKS

1. Herbert Schildt and Dale Skrien, *Java Fundamentals – A Comprehensive Introduction*, (1st ed), McGraw Hill, 2013.
2. Herbert Schildt, *Java the Complete Reference*, (7th ed), McGraw Hill, 2011.

3. T. Budd, *Understanding Object-Oriented Programming with Java,* (Updated ed), Pearson Education, 1999.

REFERENCE BOOKS

1. P. J. Dietel and H. M. Dietel, *Java How to program*, (6th ed), Prentice Hall, 2005.
2. P. Radha Krishna, *Object Oriented programming through Java,* (1st ed), CRC Press, 2007
3. S. Malhotra and S. Choudhary, *Programming in Java*, (2nd ed), Oxford University Press, 2014.

WEB LINKS:

1. https://www.codecademy.com/learn/learn-java
2. https://www.mygreatlearning.com/academy/learn-for-free/courses/java-programming
3. https://onlinecourses.nptel.ac.in/noc20_cs58/preview
4. https://www.coursera.org/specializations/object-oriented-programming

COMPUTER NETWORKS			
Semester	II	CIE Marks	50
Course Code	23MCPC523	SEE Marks	50
Teaching Hours/Week (L:T:P)	4:0:0	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. Learn the features of different computer network topologies.
2. List the required hardware to constitute a computer network.
3. Explain each computer network topology physically or logically.
4. Demonstrate error detection and correction techniques.

Module 1 : Introduction	No. of Hrs: 10
Introduction: Data Communications, Networks, The Internet, Protocols & Standards, Layered Tasks, The OSI model, Layers in OSI model, TCP/IP Protocol suite, Addressing.	
Module 2: Physical Layer-1	No. of Hrs: 12
Physical Layer-1: Analog & Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital- digital conversion (Only Line coding: Polar, Bipolar and Manchester coding), Analog-to-digital conversion (PCM, Delta), Transmission Modes, Digital-to-analog conversion.	
Module 3: Physical Layer-2 and Switching	No. of Hrs: 10
Physical Layer-2 and Switching: Multiplexing, Spread Spectrum, Introduction to switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks	
Module 4 : Data Link Layer-1	No. of Hrs: 10
Data Link Layer-1: Error Detection & Correction: Introduction, Block coding, Linear block codes, Cyclic codes, Checksum.	
Module 5: Data Link Layer-2	No. of Hrs: 10
Data Link Layer-2: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy channels, HDLC, PPP (Framing, Transition phases only), Random Access (CSMA/CD, CSMA/CA only).	

Course Outcomes:

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

CO1: Define the basic concepts of networks like protocol, internet and OSI layers.
CO2: Classify the transmission modes in a computer network.
CO3: Demonstrate the different switching techniques with its advantages and disadvantages.
CO4: Examine the functions of the Data Link Layer.

TEXT BOOKS

1. Behrouz A Forouzan, *Data Communication and Networking*, (4th ed) ,Tata McGraw-Hill, 2006.

REFERENCE BOOKS

1. Alberto Leon-Garcia, Indra Widjaja, *Communication Networks - Fundamental Concepts and Key architectures*, (2nd ed), Tata McGraw-Hill, 2004.

2. William Stallings, *Data and Computer Communication*, (8th ed), Pearson Education, 2007
3. Larry L. Peterson and Bruce S. Davie, (2007), *Computer Networks – A Systems Approach*, (4th ed), Elsevier, 2007.

WEB LINKS:

1. https://elearn.daffodilvarsity.edu.bd/course/view.php?id=5457
2. https://onlinecourses.nptel.ac.in/noc21_cs18/preview
3. https://www.youtube.com/watch?v=VwN91x5i25g
4. https://www.javatpoint.com/computer-network-tutorial

NON-RELATIONAL DATABASES			
Semester	II	CIE Marks	50
Course Code	23MCPC524	SEE Marks	50
Teaching Hours/Week (L:T:P)	4:0:0	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. Understand the concepts of Non-Relational database management systems.
2. Learn about No SQL and MongoDB.
3. Demonstrate competency in designing non-relational database management systems.

Module 1 : Introduction	No. of Hrs: 10

Introduction to NoSQL: Definition of NoSQL, History of NoSQL and Different NoSQL products. Exploring NoSQL Exploring MongoDB Java/Ruby/Python, Interfacing and Interacting with NoSQL

Module 2: NoSQL Basics	No. of Hrs: 10

NoSQL Basics: NoSQL Storage Architecture, CRUD operations with MongoDB, Querying, Modifying and Managing. Data Storage in NoSQL: NoSQL Data Stores, Indexing and ordering datasets (MongoDB/CouchDB/Cassandra)

Module 3: Advanced NoSQL	No. of Hrs: 10

Advanced NoSQL: NoSQL in Cloud, Parallel Processing with Map Reduce, Big Data with Hive.

Module 4 : Working with NoSQL	No. of Hrs: 12

Working with NoSQL: Surveying Database Internals, Migrating from RDBMS to NoSQL, Web Frameworks and NoSQL, using MySQL as a NoSQL.

Module 5: Development	No. of Hrs: 10

Developing Web Application with NOSQL and NOSQL Administration Php and MongoDB, Python and MongoDB, Creating Blog Application with PHP.

Course Outcomes:

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

CO1: Recall the concepts of structured data and unstructured data.

CO2: Explain the essential operations for creating and managing persistent data elements in relational and non-relational databases.

CO3: Categorize the differences between a traditional RDBMS and a Non-Relational database.

CO4: Characterize the steps involved in creating web applications with NoSQL.

TEXT BOOK

1. Shashank Tiwari, *Professional NOSQL*, (1st ed), WROX Press, John Wiley and Sons, Inc, 2011.

REFERENCE BOOK

1. Peter Membrey, Eelco Plugge and DUPTim Hawkins, *The Definitive Guide to MongoDB, The NoSQL Database for Cloud and Desktop Computing,* APress, (1st ed), 2010.

WEB LINKS:

1. https://www.mongodb.com/nosql-explained
2. https://youtu.be/aUPVpIYiLCc
3. https://www.coursera.org/learn/introduction-to-nosql-databases

RESEARCH METHODOLOGY & IPR			
Semester	II	CIE Marks	50
Course Code	23MCHM525	SEE Marks	50
Teaching Hours/Week (L:T:P)	4:0:0	Exam Hrs	03
Total Hours	52	Credits	04

Course Learning Objectives:

This course is designed to

1. To give an overview of the research methodology and explain the technique of defining a research problem.
2. Explain the functions of a literature review while performing a research study.
3. Learn the process of carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.
4. Make students familiar with the various research designs and their characteristics.

Module 1: Introduction to Research Methodology	No. of Hrs: 10

Research Methodology: Introduction, Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, and Problems Encountered by Researchers in India.

Module 2: Defining the Research Problem	No. of Hrs: 12

Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, An Illustration. Reviewing the literature: Place of the literature review in research, Bringing clarity and focus to your research problem, Improving research methodology, Broadening knowledge base in research area, Enabling contextual findings, How to review the literature, searching the existing literature, reviewing the selected literature, Developing a theoretical framework, Developing a conceptual framework, Writing about the literature reviewed.

Module 3: Research Design	No. of Hrs: 10

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs, Important Experimental Designs. Design of Sample Surveys: Introduction, Sample Design, Sampling and Non-sampling Errors, Sample Survey versus Census Survey, Types of Sampling Designs.

Module 4 : Data Collection	No. of Hrs: 10

Data Collection: Experimental and Surveys, Collection of Primary Data, Collection of 02.03.2021 updated 17/ 104 Secondary Data, Selection of Appropriate Method for Data Collection, Case Study Method. Interpretation and Report Writing: Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout. Types of Reports, Oral Presentation, Mechanics of Writing a Research Report, Precautions for Writing Research Reports.

Module 5 : Intellectual Property (IP) Acts	No. of Hrs: 10

Intellectual Property (IP) Acts: Introduction to IP: Introduction to Intellectual Property (IP), different types of IPs and its importance in the present scenario, Patent Acts: Indian patent acts 1970.Design Act: Industrial Design act 2000. Copyright acts: Copyright Act 1957. Trade Mark Act, 1999.

Course Outcomes:

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

- **CO1:** Identify the suitable research methods and articulate the research steps in a proper sequence for the given problem.
- **CO2:** Explain the functions of the literature review in research, carrying out a literature search, developing theoretical and conceptual frameworks and writing a review.
- **CO3:** Use data collection techniques from different sources by segregating them into primary and secondary data.
- **CO4:** Analyze some concepts/sections of CopyRight Act /Patent Act /Cyber Law/ Trademark to a given case and outline the conclusions.

TEXT BOOKS

1. C.R. Kothari, *Research Methodology, Methods and Techniques*,(4th ed), Gaurav Garg New Age International, 2018
2. Ranjit Kumar, *Research Methodology a step-by- step guide for beginners,* (3rd ed), SAGE Publications, 2011.

WEB LINKS:

1. https://www.enago.com/academy/choose-best-research-methodology/
2. https://library.tiffin.edu/researchmethodologies/whatareresearchmethodologies

Professional Electives:

AGILE SOFTWARE DEVELOPMENT			
Semester	II	CIE Marks	50
Course Code	23MCPE5261	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:0	Exam Hrs	03
Total Hours	42	Credits	03

Course Learning Objectives:

This course is designed to

1. Know the underlying concepts in agile software engineering.
2. Apply the agile design principles for software development.
3. Examine the major agile frameworks used in the current scenario.
4. Evaluate the performance of a software application with a product backlog.
5. Justify the various testing strategies for an agile software application.

Module 1 : Introduction and Project Planning	No. of Hrs: 08

Introduction: Need of agile software development, Agile context-Manifesto, Principles, Methods, Values, Roles, Artifacts, Stakeholders and Challenges– Business benefits of software agility.

Project Planning: Recognizing the structure of an agile team-Programmers, Managers, Customers, User Stories-Definition, Characteristics and Content.

Module 2: Agile Project Design	No. of Hrs: 08

Fundamentals Design Principles: Single Responsibility Approach, Open-closed principle, Liskov substitution method, Dependency – Inversion principle, Interface – Segregation.

Module 3: Common Agile Techniques	No. of Hrs: 09

Stories and backlog refinement: Agile estimation, Agile Planning, Agile testing. Agile frameworks: Major agile frameworks- Extreme programming (XP), Kanban, Feature-driven development, Lean Software Development.

Scrum Framework: Introduction to Scrum, Scrum Framework - Overview, Scrum Roles, Product Owner, Scrum Master, Development Team, Scrum Activities and Artifacts, Product Backlog, Sprints, Sprint Planning, Sprint Execution, Daily Scrum, Done, Sprint Review, Sprint Retrospective.

Module 4 : Product Backlog	No. of Hrs: 09

Product Backlog Items, Good Product Backlog Characteristics- Detailed Appropriately, Emergent, Estimated, Prioritized. Grooming- What is Grooming, Who Does the Grooming, When Does Grooming Take place? Definition of Ready. Estimation and Velocity – What and when we Estimate – Portfolio Backlog Item Estimates, Product Backlog Estimates, Task Estimates. PBI Estimation Concepts, PBI Estimation Units, Planning Poker, What is velocity, Calculate a Velocity Range, Forecasting Velocity, Affecting Velocity, and Misusing Velocity.

Module 5 : Testing	No. of Hrs: 08

The Agile lifecycle and its impact on testing, Test driven development: Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.

Course Outcomes:

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

CO1: Recognize the importance of agile software development.

CO2: Compare traditional software development and agile software development with its advantages and disadvantages.

CO3: Apply agile design principles on a software application and measure its performance.

CO4: Investigate the importance of the Scrum framework while designing agile software applications.

CO5: Evaluate the need for an agile life cycle model and its impact on software testing.

TEXT BOOKS

1. Mark Merkow, *Secure Resilient and Agile Software Development*, (1st ed), CRC Press, 2023.
2. Ken Schawber, Mike Beedle, *Agile Software Development with Scrum*, (International ed), Pearson, 2002.
3. Peter Measey, *Agile Foundations: Principles, Practices and frameworks*, (Reprint), BCS Learning & Development Limited, 2015.
4. Kenneth S. Rubin, *Essential Scrum, The Addison Wesley Signature Series*, (1st ed),Addison-Wesley and Pearson, 2012.
5. Robert C. Martin, *Agile Software Development, Principles, Patterns and Practices*, (1st ed), Prentice Hall, 2012

REFERENCE BOOKS

1. Lisa Crispin, Janet Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams* (International ed), Addison Wesley, 2009.
2. Alistair Cockburn, *Agile Software Development: The Cooperative Game*, (2nd ed) Addison-Wesley, 2006.
3. Mike Cohn, *User Stories Applied: For Agile Software*, (1st ed), Addison-Wesley, 2004.

WEB LINKS:

1. https://clearbridgemobile.com/complete-guide-agile-software development/
2. https://www.edx.org/course/agile-software-development/
3. https://www.coursera.org/learn/agile-software-development/

PRINCIPLES OF MANAGEMENT AND ORGANISATIONAL BEHAVIOR			
Semester	II	CIE Marks	50
Course Code	23MCPE5262	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:0	Exam Hrs	03
Total Hours	42	Credits	03

Course Learning Objectives:

This course is designed to

1. Understand theories and models of Management and Organisational Behavior.
2. Familiarize students with certain techniques of self-awareness.
3. Compile an adept framework for solving the problems at the workplace.
4. Acquaint the students with industry relevant skill sets.

Module 1 : Introduction	No. of Hrs: 08

Introduction: Meaning, Importance, Differences between Administration and Management, Levels of Management, Types of Managers, Managerial roles skills and competencies, Fayol's 14 principles of management, Recent trends in management.

Lab component/Activity(s): Undertake any skill development online courses on basics of Management.

Module 2: Functions of Management	No. of Hrs: 09

Planning: Process, Types of Plans, Steps in planning, Planning tools and techniques. Essentials of a good plan

Organising: Meaning, Types of Organisation structures, Span of control, Directions in organisation structures, centralisation and decentralization of authority.

Leading: Meaning, Traits and Behaviour, Contingency approaches to Leadership, Transformational leadership. **Controlling**: Meaning, Importance, Steps in the control process, Resistance to control, Types of Control, Control techniques

Lab component/Activity(s): Conduct an event in the department and try to understand the various roles played by students in relation to Team and Organisational environment.

Module 3: Organisational Behavior	No. of Hrs: 09

Organisational Behavior: Meaning, Approaches to organisational behavior, models of Organisational behavior.

Behavioural Dynamics: MARS Model of individual behavior and performance, Types of Individual behavior, Personality in Organisation, **Values:** Values at the workplace, Types of values, **Perception:** Meaning, Model of Perceptual process. Factors influencing perception, Perception and decision making. **Emotions**: Types of emotions, Circumflex Model of Emotion, Stress and its management **Attitudes:** Meaning, Types. Attitudes and behavior, changing attitudes.

Lab component/Activity(s): Develop questions, and try to observe personality traits of the self.

Module 4 : Personality	No. of Hrs: 08

Personality: Definition, factors influencing personality, Big Five personality traits, Myers-Briggs personality Indicator (MBTI), Personality tools and tests, **Motivation:** Definition, Process of motivation, Cycle of motivation, Types, theories – Maslow's Hierarchy of needs, Four drive theory of motivation.

Lab component/Activity(s): Conduct self introspection by applying four drive theories of motivation.

Module 5: Group Dynamics	No. of Hrs: 08

Group Dynamics: Meaning, Group characteristics, Classification of groups, Models of group development, meaning of group dynamics **Teams:** Meaning, Team characteristics, Teams v/s groups, Model of Team Effectiveness, Stages of Team Development. Creating effective teams.

Lab component/Activity(s): Identify a team and analyze the various stages of development.

Course Outcomes:

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

CO1: Recognize the different theories in the field of Management.

CO2: Discuss management and behavioral models related to values, perceptions, emotions and attitudes to solve business problems.

CO3: Use the recent concepts in understanding personality and motivation towards effective team building.

TEXT BOOKS

1. Chandrani Singh and Aditi Khatri, *Principles and Practices of Management and Organizational Behavior*, (1st ed)SAGE publication, 2016.
2. Koontz, *Essentials of Management*, (8th ed), McGraw Hill, 2014.

REFERENCE BOOKS

1. Stephen P Robins, Timothy, *Organizational Behavior* (14th ed), Pearson, 2012.
2. Chuck Williams & Manas Ranjan Tripathy, *MGMT: A South-Asian Perspective* (5th ed), Cengage Learning, 2013.
3. Fred Luthans, *Organizational Behavior*, (12th ed), McGraw Hill International, 2011.
4. John R. Schermerhorn, *Management*, (8th ed), Wiley India, 2010.
5. Ramesh B Rudani, *Principles of Management*, Tata McGraw-Hill, 2013.

WEB LINKS:

1. https://onlinecourses.nptel.ac.in/noc22_mg104/preview
2. https://onlinecourses.nptel.ac.in/noc22_mg78/preview

CRYPTOGRAPHY AND NETWORK SECURITY			
Semester	II	CIE Marks	50
Course Code	23MCPE5263	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:0	Exam Hrs	03
Total Hours	42	Credits	03

Course Learning Objectives:

This course is designed to

1. Learn the basics of network security.
2. Practice the different encryption techniques.
3. Gain knowledge on hash functions, MAC and their use in various protocols of network security.
4. Illustrate the importance of digital signatures.

Module 1: Introduction	No. of Hrs:08

Introduction: Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security. Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography.

Module 2: Block Ciphers and The Data Encryption Standard	No. of Hrs: 09

Block Ciphers and The Data Encryption Standard: Block Cipher Principles, The Data Encryption Standard (DES), The Strength of DES, Block Cipher Design Principles. Block Cipher Operation: Multiple Encryption and Triple DES, Electronic Codebook Mode, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode. Stream Ciphers: Stream Ciphers, RC4.

Module 3: Number Theory	No. of Hrs: 09

Number Theory: Divisibility and the Division Algorithm, The Euclidean Algorithm, Modular Arithmetic, Prime Numbers, Fermat's and Euler's Theorems, Testing for Primality. Public-Key Cryptography, RSA And Other Public-Key Cryptosystems: Principles of Public-Key Cryptosystems, The RSA Algorithm, Diffie-Hellman Key Exchange.

Module 4: Cryptographic Hash Functions	No. of Hrs: 08

Cryptographic Hash Functions: Applications of Cryptographic Hash Function, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA). Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Message Authentication Codes, MACs Based on Hash Functions (HMAC).

Module 5: Digital Signature	No. of Hrs: 08

Digital Signatures: Digital Signatures, Schnorr Digital Signature Scheme, Digital Signature Standard (DSS). Key Management And Distribution: Symmetric Key Distribution Using Symmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public Key Infrastructure.

Course Outcomes:

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

CO1: Define classical encryption techniques and block ciphers in today's network scenarios.

CO2: Explain the broad steps in the data encryption standard with its benefits.

CO3: Illustrate the different public-key cryptography schemes, RSA and other public-key

cryptosystems.

CO4: Examine key management and distribution schemes and design user authentication, such as Diffie-Hellman Key Exchange.

TEXT BOOKS

1. William Stallings, *Cryptography And Network Security - Principles And Practice*, (5th ed), Pearson/PHI, 2011.

REFERENCE BOOKS

1. William Stallings, *Network Security Essentials (Applications and Standards)*, (4th ed), Pearson Education, 2012.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, *Network Security – Private Communication in a Public World*, (2nd ed), Pearson/PHI, 2002.
3. Eric Maiwald, *Fundamentals of Network Security*, (1st ed), Dreamtech Press, 2003.
4. Whitman, *Principles of Information Security*, (3rd ed), Thomson, 2009.

WEB LINKS:

1. https://www.youtube.com/watch?v=JoeiLuFNBc4
2. https://onlinecourses.nptel.ac.in/noc21_cs16/preview
3. https://www.udemy.com/course/du-cryptography/
4. https://www.coursera.org/learn/crypto

COMPUTER GRAPHICS WITH OPENGL			
Semester	II	CIE Marks	50
Course Code	23MCPE5264	SEE Marks	50
Teaching Hours/Week (L:T:P)	3:0:0	Exam Hrs	03
Total Hours	42	Credits	03

Course Learning Objectives:

This course is designed to

1. Enumerate the key concepts of graphics output primitives and attributes with OpenGL.
2. Describe the algorithms and theories that form the basis of computer graphics.
3. Demonstrate the production of 2D and 3D transformations.
4. Apply concepts of clipping and visible surface detection in 2D and 3D viewing, Illumination Models.
5. Create curves and computer-animated images using OpenGL.
6. Decide suitable hardware and software for developing graphics packages using OpenGL.

Module 1: Graphics Output Primitives and Attributes	No. of Hrs: 08

Introduction to OpenGL, Coordinate reference frames, Specifying two dimensional world coordinate reference frame in OpenGL, OpenGL point functions, OpenGL line functions, Line drawing algorithms, Circle generation algorithms, Ellipse generation algorithms, Fill area primitives, Polygon fill areas, OpenGL polygon fill area functions, General scan line polygon fill algorithm, Fill methods for areas with irregular boundaries, OpenGL fill area attribute functions

Module 2: Two – Dimensional and Three - Dimensional Geometric Transformations	No. of Hrs: 08

Basic two dimensional geometric transformations, Matrix representations and homogeneous coordinates, Inverse transformations, Two dimensional composite transformations, Other two dimensional transformations, Three dimensional Translation, Rotation, Scaling, Other three dimensional transformations, Affine transformations, OpenGL geometric transformation functions.

Module 3: Two Dimensional Viewing	No. of Hrs: 08

The two dimensional viewing, Clipping window, Normalisation and viewport transformations, Clipping algorithms, Two dimensional point clipping, Two dimensional line clipping algorithms, Polygon fill area clipping, Curve clipping, Text clipping.

Module 4: Three Dimensional Viewing	No. of Hrs: 09

The three dimensional viewing concepts, Three dimensional viewing pipeline, Three dimensional viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformations, Orthogonal projections, Oblique parallel projections, Perspective projections, The viewport transformation and three dimensional screen coordinates.

Module 5: Curves and Computer Animation	No. of Hrs: 09

Bezier spline curves, Raster methods for computer animation, Design of animation sequences, Traditional animation techniques, General computer animation functions.

Course Outcomes:

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

CO1: Describe the key concepts of output primitives and attributes with OpenGL.

CO2: Discuss the steps involved in creating graphics objects using geometric transformations and their applications.

CO3: Illustrate scene generation using different clipping methods with their transformation.

CO4: Investigate the different visible surface detection techniques for display of 3D scenes.

TEXT BOOKS

1. Donald Hearn, M.Pauline Baker, *Computer Graphics with OpenGL*, (3rd ed), Indian Edition, Pearson, 2004.

REFERENCE BOOKS

1. Edward Angel, *Interactive Computer Graphics* – A top down approach using OpenGL, (6th ed), Pearson, 2011.
2. Peter Shirley, Steve Marschner, *Fundamentals of Computer Graphics*, (3rd ed), Cengage Learning Indian edition, 2009.

WEB LINKS:

1. https://www.coursera.org/learn/interactive-computer-graphics
2. https://onlinecourses.nptel.ac.in/noc21_cs97/preview
3. https://www.edx.org/learn/computer-graphics/the-university-of-california-san-diego-computer-graphics

MINI-PROJECT			
Semester	II	CIE Marks	50
Course Code	23MCSE527	SEE Marks	50
Teaching Hours/Week (L:T:P)	0:0:2	Exam Hrs	2.5
Total Hours	26	Credits	02

Course Learning Objectives:

This course is designed to

1. Recall the concepts learnt in database management system courses.
2. Apply the required tools and techniques for software development.
3. Examine the requirements and transform them to a software module.
4. Assess the valid arguments in case study against the software module developed.
5. Formulate the test cases and strategies for the software module developed.

The Mini Project is based on implementation of concepts and theory learnt in programming languages and DBMS. The sample project titles are listed as follows.

1. Barcode Generation
2. Bank software with ATM
3. Load shedding in mobile systems
4. Document security system
5. Project planning and management
6. Library members information system
7. College Enrolment system
8. Resilient online coverage for surveillance applications
9. Employee information and payroll system
10. Any other application or system

Guidelines:

- Project must be done individually.
- Final evaluation will be done through project demonstration.
- The marks of the mini project would be given on the basis of performance in CIE and SEE.

Evaluation:

During project work, the evaluation process will be divided into a number of phases to assess the continuous progress (Minimum three phases).

- The project guides and project coordinator follows rubrics, which is set by the Department for evaluation and then submitted to the head of department.
- Each internal guide will verify the statement of project and literature of works and implementation details. The department will encourage students to make publications in standard conferences/journals.

Rubrics for Mini Project Evaluation CIE & SEE:

	Review #	Agenda	Assessment	Review Assessment Weightage	Overall Weightage	
	Review 1	Project Synopsis Evaluation	Rubrics 1	25	25 (Avg of R1, R2, R3)	
	Review 2	Mid-Term Project Evaluation	Rubrics 2	25		
	Review 3	Final Internal Project Evaluation	Rubrics 3	25		
	Final Project Viva-Voce	End-Semester Project Evaluation		25	25	
	Total				50	

WEB LINKS:

- https://www.youtube.com/watch?v=-GwBNwZOPUs
- https://www.youtube.com/watch?v=9PgZCJNzY9M