AUTONOMOUS

SYLLABUS

2025

DEPARTMENT
OF
MASTER OF COMPUTER APPLICATIONS

MITE

Invent Solutions

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

Institute 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**"*

Institute 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 and Practical, Theoretical and Pragmatic;** ALL at the same time"*

Department Vision

To attain global recognition as a leader in imparting quality education in the field of computer applications by fostering an inclusive and dynamic learning environment that nurtures research skills, creativity, innovation, and critical thinking ability of the students.

Department Mission

- *To deliver modern multi-disciplinary technological skills in the field of computer applications.*
- *To cultivate a research culture among the faculty and students to the betterment of the human society at large.*
- *To make students cognizant of the importance of ethical and moral values while creating futuristic software applications.*

Program Educational Objectives (PEOs)

- *Design effective computing systems that are technically viable and fulfilling the requirements of customers.*
- *Develop a professional career to contribute towards interdisciplinary academic and research projects.*
- *Acquire and apply the knowledge of existing tools and technologies from time to time to industry-oriented and societal problems.*

Program Specific Outcomes (PSOs)

- *Apply the principles of Software Engineering and Project management to design, develop and maintain software applications using Programming languages and frameworks.*
- *Apply computational techniques and algorithms to solve real-world problems in artificial intelligence and cybersecurity environments.*

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

LIST OF COURSES

I/II SEMESTER			
Sl. No	**Course Code**	**Course Title**	**Sem**
PROFESSIONAL CORE (PC)			
1	25MCPC501	Mathematical Foundation for Computer Applications	**I**
2	25MCPC502	Problem Solving and Programming in Python	**I**
3	25MCPC503	Operating System	**I**
4	25MCPC504	Data Structures	**I**
5	25MCPC505	Computer Networks	**I**
6	25MCPC506	Software Engineering	**I**
7	25MCPC507	Web Technologies	**I**
8	25MCPC511	Design and Analysis of Algorithms	**II**
9	25MCPC512	Programming in Java	**II**
10	25MCPC513	Database Management Systems	**II**
11	25MCPC514	Object Oriented Modeling and Design	**II**
12	25MCPC601	Cloud Computing	**III**
13	25MCPC602	Advanced Java	**III**
PROFESSIONAL ELECTIVE (PE)			
14	25MCPE521	Cryptography and Cyber Security	**II**
15	25MCPE522	Network Security	**II**
16	25MCPE523	Machine Learning	**II**
17	25MCPE524	Pattern Recognition	**II**
18	25MCPE621	Ethical Hacking & Secure Coding	**III**
19	25MCPE622	Blockchain Technology And Cyber Intelligence	**III**
20	25MCPE623	Deep Learning for Natural Language Processing	**III**
21	25MCPE624	Computer Vision for Image and Video Analytics	**III**
22	25MCSE612	MOOC	**IV**
HUMANITIES (HM)			
23	25MCNM508	Professional Communication and Ethics	**I**
24	25MCHM515	Research Methodology and IPR	**II**
SKILL ENHANCEMENT (SE)			
25	25MCSE516	Mini Project	**II**
26	25MCSE603	Major Project	**III**
27	25MCSE611	Internship	**IV**
28	25MCSE613	Publication/ Patenting	**IV**

I SEMESTER

Sl. No	Course Code	Course Title	Category	Teaching Dept.	Teaching Hours / Week			Exam Marks			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPC501	Mathematical Foundation for Computer Applications	Professional Core	Mathematics	2	2	0	50	50	100	3	3
2	25MCPC502	Problem Solving and Programming in Python	Professional Core	MCA	2	0	2	50	50	100	3	3
3	25MCPC503	Operating System	Professional Core	MCA	3	0	0	50	50	100	3	3
4	25MCPC504	Data Structures	Professional Core	MCA	2	0	2	50	50	100	3	3
5	25MCPC505	Computer Networks	Professional Core	MCA	3	0	0	50	50	100	3	3
6	25MCPC506	Software Engineering	Professional Core	MCA	3	0	0	50	50	100	3	3
7	25MCPC507	Web Technologies	Professional Core	MCA	2	0	2	50	50	100	3	3
8	25MCNM508	Professional Communication and Ethics	Humanities	Humanities /MCA	2	0	0	100	-	100	-	-
										Total		21

II SEMESTER

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPC511	Design and Analysis of Algorithms	Professional Core	MCA	4	0	0	50	50	100	3	4
2	25MCPC512	Programming in Java	Professional Core	MCA	3	0	2	50	50	100	3	4
3	25MCPC513	Database Management Systems	Professional Core	MCA	3	0	2	50	50	100	3	4
4	25MCPC514	Object Oriented Modeling and Design	Professional Core	MCA	2	0	2	50	50	100	3	3
5	25MCHM515	Research Methodology and IPR	Humanities	MCA/ Humanities	3	0	0	50	50	100	3	3
6	25MCPE52X	Professional Elective - 1	Professional Elective	MCA	**	**	**	50	50	100	3	3
7	25MCSE516	Mini Project	Skill Enhancement	MCA	-	-	4	50	50	100	2.5	2
Total Credits												23

**Refer to the following table

Professional Elective – 1
Stream I – Cyber Security

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Exam Marks			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPE521	Cryptography and Cyber Security	Professional Elective	MCA	3	0	0	50	50	100	3	3
2	25MCPE522	Network Security	Professional Elective	MCA	3	0	0	50	50	100	3	3

Stream II - AI & ML

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Exam Marks			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPE523	Machine Learning	Professional Elective	MCA	2	0	2	50	50	100	3	3
2	25MCPE524	Pattern Recognition	Professional Elective	MCA	2	0	2	50	50	100	3	3

III SEMESTER

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPC601	Cloud Computing	Professional Core	MCA	3	0	0	50	50	100	3	3
2	25MCPC602	Advanced Java	Professional Core	MCA	3	0	2	50	50	100	3	4
3	25MCPE62X	Professional Elective - 2	Professional Elective	MCA	3	0	2	50	50	100	3	4
5	25MCSE603	Major Project	Skill Enhancement	MCA	-	-	-	100	-	100	-	8
Total Credits												19

Professional Elective – 2

Stream I – Cyber Security

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPE621	Ethical Hacking & Secure Coding	Professional Elective	MCA	3	0	2	50	50	100	3	4
2	25MCPE622	Blockchain Technology And Cyber Intelligence	Professional Elective	MCA	3	0	2	50	50	100	3	4

Stream II - AI & ML

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCPE623	Deep Learning for Natural Language Processing	Professional Elective	MCA	3	0	2	50	50	100	3	4
2	25MCPE624	Computer Vision for Image and Video Analytics	Professional Elective	MCA	3	0	2	50	50	100	3	4

IV SEMESTER

Sl. No	Course Code	Course Title	Category	Teaching Dept	Teaching Hours / Week			Examination (Marks)			Duration of Exam (SEE)	Credits
					L	T	P	CIE	SEE	Total		
1	25MCSE611	Internship	Skill Enhancement	-	-	-	-	50	50	100	3	10
2	25MCSE612	MOOC*	Professional Elective	-	-	-	-	*	*	-	-	3
3	25MCSE613	Publication/ Patenting	Skill Enhancement	-	-	-	-	100	-	100		4
											Total Credits	17

***Mandatory NPTEL Course**

*** CIE (50 Marks), * SEE (50 Marks)**

I SEM

MATHEMATICAL FOUNDATION FOR COMPUTER APPLICATIONS			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC501**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:2:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Lay 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. Compute and interpret correlation coefficients to measure the relationship between two variables and apply simple linear regression to predict one variable based on another.
5. Build a strong foundation in probability theory to solve problems involving random phenomena.

Module 1: Sets, Relations and Functions	**No. of Hours: 08**

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, Applications of sets, relations 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.
Textbook 1 - Ch 2

Pedagogy	Chalk-talk, ppt

Module 2: Mathematical logic	**No. of Hours: 10**

Basic connectivity and Truth table, Logical equivalences, Predicative Logic, Rules of inference, Quantifiers
Self-study: Performing logical operations using MATLAB.
Textbook 1 - Ch 1

Pedagogy	Chalk-talk, ppt

Module 3: Graph theory	**No. of Hours: 08**

Graphs & graph models, Subgraphs, Complement and Graph Isomorphism, Connectivity, Euler's & Hamiltonian paths, Graph theory applications in solving real-life problems.
Self-study: Representing graphs, and determining its combinatorial properties using MATLAB
Textbook 1 - Ch 9

Pedagogy	Chalk-talk, ppt

Module 4: Correlation & Regression	**No. of Hours: 08**

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. **Textbook 2 - Ch 11**	
Pedagogy	Chalk-talk, ppt
Module 5: Probability	**No. of Hours: 08**
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 **Textbook 2 - Ch 2**	
Pedagogy	Chalk-talk, ppt

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

Textbooks:

1. Kenneth H Rosen, "*Discrete Mathematics and its Applications"*, 8th Edition, Tata McGraw-Hill Education Private Limited, 2023
2. Ronald E. Walpole, Sharon L Myers, "*Probability and Statistics for engineers and Scientist"*, 9th Edition, Pearson Education, 2022

Reference Books:

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

Web Links:

1. Introduction to theory of Sets-https://nptel.ac.in/courses/111107058
2. Mathematical Logics-https://onlinecourses.nptel.ac.in/noc20_cs82/preview
3. Introduction: Regression Analysis-NPTEL :: Mathematics - Regression Analysis

PROBLEM SOLVING AND PROGRAMMING IN PYTHON			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC502**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:
This course is designed to
1. Impart knowledge on the different collection objects and functions in Python.
2. Provide skills to use object-oriented programming constructs using Python.
3. Develop skills to use NumPy array functionalities and pandas data structures for data analysis.
4. Realize data visualization skills using Matplotlib and Seaborn libraries.

Module 1: Python Basic Concepts and Programming — **No. of Hours: 06+06**

Using Python Interpreter, Introduction to Python, Running Python, The First Program, Arithmetic Operators, Values and Types, Formal and Natural Languages, Variables, Expressions and Statements, Functions, Conditionals and Recursion, Fruitful functions, Iteration, More Control Flow tools.
Textbook 2 – Ch 2, 3, 4
Textbook 1 – Ch 1, 2, 3, 4, 5, 6, 7

Laboratory Components:
Programs should be written and executed for the following:
1. Performing a linear search on a given dataset
2. Insertion of an element into a sorted list.
3. Use Object Oriented programming to show encapsulation, overloading and inheritance.

Module 2: Python Collection Objects, Classes — **No. of Hours:05+06**

Strings, Lists, Dictionaries, Tuples, Files, Classes and Objects, Classes and Functions, Classes and Methods, Inheritance.
Textbook 1 – Ch 8, 10, 11, 12, 14, 15, 16, 17, 18
Textbook 2 – Ch 5, 9

Laboratory Components:
4. Use pandas to perform a) Importing Datasets b) Cleaning the Data c) Data frame manipulation
5. Use NumPy to perform a) Array Manipulation, Searching, Sorting and Splitting b) Broadcasting and Plotting NumPy arrays.
6. Use NumPy and Matplotlib for data visualization of various types of graphs.

Module 3: Introduction to NumPy and pandas — **No. of Hours: 05+06**

Numpy: Understanding datatypes in Python, Basics of NumPy arrays, Computation on NumPy arrays: Universal Function.
Pandas: Introduction to pandas Data Structure, Essential Functionally, Summarizing and Computing Descriptive Statistics, Handling Missing Data.
Textbook 3 – Ch 2, 3

Laboratory Components:
7. Use NumPy to create an m x n integer array, print its attributes and use Matplotlib to visualize it Generate a) Linear regression models b) Logistic regression models.

Module 4: Data Loading and Data Wrangling	**No. of Hours: 05+06**

Reading and Writing data in text format, Interacting with the Database, Combining and Merging datasets, Reshaping and Pivoting, Data Transformation, String Manipulation.
Textbook 3 – Ch 3

Laboratory Components:

8. Time Series analysis with pandas.
9. Data visualization using Seaborn

Module 5: Visualization with Matplotlib and Seaborn	**No. of Hours: 05**

General Matplotlib tips, Simple Line Plots, Simple Scatter Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binning and Density, Customizing Plot Legends and Color Bars, Customizing Matplotlib, Visualization with Seaborn.
Textbook 3 – Ch 4

Course Outcomes:

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

CO1: Use Python programming concepts like flow control and decision-making constructs to solve problems.
CO2: Utilize the concepts of lists, dictionaries and strings and its associated functions in Python programs to write programs for a given problem definition.
CO3: Make use of NumPy array functionalities and pandas framework for data manipulations.
CO4: Apply the data transformation techniques on different datasets and arrive at data-driven insights.
CO5: Utilize data visualization tools such as Matplotlib, Seaborn to create various types of data visualizations.

Textbooks:

1. Allen B. Downey, "*Think Python: How to Think Like a Computer Scientist*", 2nd Edition, Shroff/O'Reilly Publishers, 2016
2. Guido Van Rossum, Fred L. Drake Jr, "*An Introduction to Python*", Network Theory Ltd, 2011
3. Jake Vander Plas, "*Python Data Science Handbook: Essential tools for Working with Data*", 1st Edition, O'Reilly Publishers, 2016

Reference Books:

1. Mark Lutz, "*Programming Python: Powerful Object-Oriented Programming*", 4th Edition, Shroff/O'Reilly Publishers, 2011
2. Tim Hall, J-P Stacey, "*Python 3 for Absolute Beginners*", 1st Edition, Apress, 2009
3. Magnus Lie Hetland, "*Beginning Python: From Novice to Professional*", 2nd Edition, Apress, 2009
4. Shai Vaingast, "*Beginning Python Visualization: Crafting Visual Transformation Scripts*", 1st Edition, Apress, 2009

Web Links:

1. Programming in Python: https://onlinecourses.swayam2.ac.in/cec24_cs11/preview
2. Python for Data Science: https://onlinecourses.nptel.ac.in/noc24_cs68/preview
3. Python for Everybody: https://www.coursera.org/specializations/python

OPERATING SYSTEM			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC503**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Familiarize the basic concepts and structure of operating systems.
2. Impart knowledge on different process scheduling techniques.
3. Familiarize with the basics of process management, deadlocks and file systems.
4. Provide skills on using the Unix commands on the Unix/Linux operating system.
5. Impart the basic knowledge of shell programming on the Unix operating system.

Module 1: Introduction	**No. of Hours: 08**

Overview: Introduction to Operating Systems, Computer System Architecture, Operating System Operations, Distributed Systems, Open-source operating systems.

System Structures: Operating System Services, System Calls, Types of System Calls, System Services, Operating System Structure, Virtualization.

Textbook 1 - Ch 1, 2

Module 2: Process Management	**No. of Hours: 08**

Process Management: Process concept, process state, process control block, Process Scheduling: Basic concepts, Scheduling criteria, Scheduling Algorithms: FCFS, SJFS, Priority scheduling, Round Robin Scheduling. **Multithreaded Programming:** Overview, Multithreaded Models, Threading Issues.

Textbook 1 - Ch 3, 4, 5

Module 3: Synchronization and Deadlocks	**No. of Hours: 08**

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

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

Textbook 1 - Ch 6, 7, 8

Module 4: The File System	**No. of Hours: 09**

The File System: The File & File name, Unix File System, The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames.

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.

Textbook 2 - Ch 5, 6, 11

Module 5: Shell Programming	**No. of Hours: 09**

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.

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.

Textbook 2 - Ch 8, 14

Course Outcomes:

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

CO1: Outline the components in a typical operating system structure and use functions, system calls to perform different operations.

CO2: Utilize the different scheduling algorithms and solve problems in concurrency and deadlock management.

CO3: Apply appropriate techniques to solve process synchronization problems and implement efficient thread handling mechanisms in a multithreaded environment.

CO4: Identify the appropriate Linux commands for memory management, file management, and directory management tasks.

CO5: Develop shell scripts to automate routine tasks and manage system operations effectively.

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "*Operating Systems Concepts",* 10th Edition, Wiley India, 2018.
2. Sumitabha Das, "*UNIX Concepts and Applications",* 4th Edition, Tata McGraw Hill, 2006.

Reference Books:

1. D. M. Dhamdhere, "*Operating Systems – A Concept Based Approach",* 2nd Edition, Tata McGraw – Hill, 2006
2. P. C. P. Bhatt, "*Operating Systems"*, 2nd Edition, PHI, 2006
3. W. Richard Stevens Stephen A. Rago, "*Advanced Programming in the UNIX Environment"*, 3rd Edition, Addison Wesley, 2013

Web Links:

1. Operating Systems Fundamentals: https://www.coursera.org/learn/akamai-operating-systems
2. Operating System: https://onlinecourses.nptel.ac.in/noc20_cs04/preview
3. Operating Systems Course for Beginners: https://www.youtube.com/watch?v=yK1uBHPdp30
4. Basic Unix Commands: https://www.unixtutorial.org/basic-unix-commands
5. Shell Scripting Tutorial: https://www.youtube.com/watch?v=GtovwKDemnI

DATA STRUCTURES			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC504**	SEE Marks	**50**
Teaching Hours/Week(L:T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory+24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Familiarize the basic concepts of data structures.
2. Impart knowledge on the concepts of stack and its operations.
3. Develop an understanding of various types of queues and their implementations.
4. Construct an efficient program using appropriate data structures.
5. Develop a deep understanding of various tree and graph types.

Module 1: Classification of Data Structures | **No. of Hours: 06+06**

Introduction to data structure, Types of data structures, Data structure Operations, Arrays: Linear arrays, Representation of Linear array in memory, Traversing linear array, Sorting Techniques: Bubble Sort, Selection Sort, Merge sort, Quicksort, Searching: Sequential search, Binary search.

Textbook 3 - Ch 3, 4

Textbook 2 - Ch 4, 6

Laboratory Component

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

Module 2: Stack and its Applications | **No. of Hours: 05+06**

Stack: Definition, Representation, Operations and Applications: Polish and Reverse polish expressions, Conversion of expression, evaluation of postfix expression, Recursion: Factorial, Fibonacci Sequence.

Textbook 3 - Ch 6

Laboratory Component

3. Write a C program to implement stack with the following operations:
   a) Push an element on to stack.
   b) Pop an element from the stack.
4. Write a C program to perform the following using recursion:
   a) Find the Factorial of a given number.
   b) Generate the Fibonacci series up to *n* terms.

Module 3: Queue and its Applications	**No. of Hours: 05+04**

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

Textbook 1 - Ch 3

Laboratory Component

5. Design, develop, and execute a program in C to simulate the working of a simple queue of integers using an array. Perform the following operations:
   a. Insert
   b. Delete
   c. Display

Module 4: Linked List	**No. of Hours: 05+04**

Limitations of array implementation, Memory Management: Static (Stack) and Dynamic (Heap) Memory Allocation, Memory management functions. Linked list types: Singly Linked List. Linked list as a Data Structure, Inserting and removing nodes from singly linked list, Linked implementations of stacks and queues.

Textbook 3 - Ch 7

Laboratory Component

6. Write a C program to simulate the working of a singly linked list with the following operations:
   a) Insert at the beginning
   b) Insert at the end
   c) Insert at a given position
   d) Display

Module 5: Trees and Graph	**No. of Hours: 05+04**

Terminology, Binary Trees, Properties of Binary trees, Binary Search Tree, 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, Operations, Traversal methods: Breadth First Search and Depth First Search.

Textbook 3 - Ch 8, 9

Textbook 1 - Ch 5, 6

Laboratory Component

7. 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).

Course Outcomes:

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

CO1: Apply appropriate data structures and algorithms such as arrays, sorting, and searching techniques to solve computational problems efficiently.

CO2: Apply stack operations and recursive techniques to evaluate expressions and solve computational problems such as factorial and Fibonacci sequence generation.

CO3: Make use of the different queue structures and their operations to develop solutions for real-time

problems such as process scheduling and resource management.

CO4: Apply dynamic memory allocation and linked list concepts to implement efficient data structures such as stacks and queues.

CO5: Use tree and graph traversal techniques to represent, process, and solve problems using appropriate data structures such as binary trees and graphs.

Textbooks:

1. Ellis Horowitz and Sartaj Sahni, "*Fundamentals of Data Structures in C",* 2nd Edition, Universities Press, 2023
2. Seymour Lipschutz, "*Data Structures Schaum's Outlines",* Revised 1st Edition, McGraw Hill, 2014
3. Bala Guruswamy, "*Programming in ANSI C"*, 8th Edition, McGraw Hill, 2024

Reference Books:

1. Gilberg & Forouzan, "*Data Structures: A Pseudocode approach with C",* 2nd Edition, CengageLearning, 2014
2. Reema Thareja, "*Data Structures using C"*, 3rd Edition, Oxford Press, 2012

Web Links:

1. Introduction to Data Structure: https://archive.nptel.ac.in/courses/106/102/106102064/
2. Trees and Graphs:www.javatpoint.com/tree-vs-graph-data-structure3
3. Binary Tree:https://www.mygreatlearning.com/blog/data-structures-using-c/#binary-tree

COMPUTER NETWORKS			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC505**	SEE Marks	**50**
Teaching Hours/Week(L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Build an understanding of the fundamental concepts in computer networks.
2. Outline the different challenges in communication and to provide appropriate solutions.
3. Summarize the limitations of the different layers in the OSI reference model while transmitting data.
4. Demonstrate error detection and correction techniques.

Module 1: Introduction	**No. of Hours: 10**

Uses of Computer Networks, Types of computer network, Network Technology from local to global, Reference Models: The OSI Reference Model, The TCP/IP Reference Model, A Comparison of the OSI and TCP/IP Reference Models, Networks protocols, Protocol Layering.
Physical Layer- Guided Transmission Media, Wireless Transmission.
Textbook 1- Ch 1, 2

Module 2: Data Link Layer	**No. of Hours: 08**

Data link Layer design issues, Error Detection and correction Techniques, Elementary data link protocol, Simple Protocol, Stop and Wait Protocol, Stop and Wait ARQ Protocol, Sliding Window Protocols (Go-Back-N (GBN) and Selective Repeat (SR)).
Textbook 1 - Ch 3

Module 3: Network Layer	**No. of Hours: 08**

Network Layer Design issues, Routing algorithms- Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing, Hierarchical routing, IPV4 Protocol, IP Addresses, IPV6 Protocol.
Textbook 1 - Ch 5

Module 4: Transport Layer	**No. of Hours: 08**

Transport-Layer Services, Elements of transport protocol: Addressing, Connection establishment and Release, Error control and Flow control, Multiplexing, Crash Recovery, The Internet Transport Protocol: UDP, The Internet Transport Protocol: TCP
Textbook 1 - Ch 6

Module 5: Application Layer	**No. of Hours: 08**

Introduction, Client Server Programming, WWW and HTTP, FTP, Electronic Mail, Telnet, Domain Name System (DNS), Multimedia Data, Multimedia in the internet.
Textbook 2 - Ch 25, 26, 28

Course Outcomes:

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

CO1: Outline the fundamental principles of computer networking and enumerate its layers and protocols.

CO2: Identify the issues related to design, services, interfaces and protocols in the data link layer and network layer.

CO3: Apply various routing algorithms to solve a given problem in the network layer.

CO4: Make use of transport-layer protocols to design and analyze reliable data communication using TCP and UDP.

CO5: Apply client-server concepts and internet protocols to develop and analyze web-based multimedia applications.

Textbooks:

1. Andrew S. Tanenbaum, David J Wetherall, "*Computer Networks"*, 6th Edition, Pearson Education, 2023
2. Behrouz A Forouzan, "*Data Communication and Networking"*, 5th Edition, Tata McGraw-Hill, 2013

Reference Books:

1. Alberto Leon-Garcia, Indra Widjaja, "*Communication Networks - Fundamental Concepts and Key architectures",* 2nd Edition, Tata McGraw-Hill, 2004
2. William Stallings, "*Data and Computer Communication",* 8th Edition, Pearson Education, 2007

Web Links:

1. Introduction to Computer Network: https://www.javatpoint.com/computer-network-tutorial
2. Data Communication: https://elearn.daffodilvarsity.edu.bd/course/view.php?id=5457
3. Computer Networks and Internet Protocol: https://onlinecourses.nptel.ac.in/noc21_cs18/preview
4. Error Detection in Computer Networks: https://www.geeksforgeeks.org/error-detection-in-computer- networks/

SOFTWARE ENGINEERING			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC506**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Impart the fundamental principles, methodologies, and best practices involved in software engineering.
2. Learn to gather, analyze, and document both functional and non-functional requirements from stakeholders to ensure the software meets user needs.
3. Enable students to design scalable, maintainable, and efficient software systems using appropriate software design patterns and architectural styles (e.g., MVC, layered architecture, microservices).
4. Provide knowledge on the various testing techniques unit, integration, system, and acceptance testing to ensure software quality.
5. Impart skills on effective project management and quality management.

Module 1: Introduction	**No. of Hours: 08**

Professional software development: Software engineering ethics, **Software processes:** Software process models, Process activities, Coping with change, The rational unified process.
Textbook 1 - Ch 1, 2

Module 2: Requirements Engineering	**No. of Hours: 08**

Requirements engineering: Functional and non-functional requirements, The Software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management.
Textbook 1 - Ch 4

Module 3: Software Architecture	**No. of Hours: 08**

Role of Software Architecture, Architectural Views, **Component and Connector View: Components, Connectors, Architectural Styles for C & C View:** Pipe and Filter, Shared-Data Style, Client-Server Style, Other Styles, **Architectural Discussion:** Architecture and Design, Preserving the Integrity of an Architecture, Deployment View and Performance Analysis, Documenting Architecture Design, **Evaluating Architectures:** The ATAM Analysis Method.
Textbook 2 - Ch 4

Module 4: Software Testing	**No. of Hours: 12**

Testing Fundamentals : Error, Fault, and Failure, Test Oracles, Test Cases and Test Criteria, Psychology of Testing**, Black-box Testing** : Equivalence Class Partitioning, Boundary Value Analysis, Cause-Effect Graphing, Pair-wise Testing, Special Cases, State-Based Testing**, White-Box Testing :** Control Flow-Based Criteria, Data Flow-Based Testing, Mutation Testing, Test Case Generation and Tool Support, **Testing Process :** Levels of Testing, Test Plan, Test Case Specifications, Test Case Execution and Analysis, Defect Logging and Tracking, **Defect Analysis and Prevention :** Pareto Analysis, Perform Causal Analysis, Develop and Implement Solutions.
Textbook 2 - Ch 10

Module 5: Project Management	**No. of Hours: 06**

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.
Textbook 1 - Ch 22, 23, 24

Course Outcomes:

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

CO1: Apply the right software process model for a given problem scenario.
CO2: Identify the functional and non-functional requirements while building a software application.
CO3: Choose an appropriate architectural style for a given problem.
CO4: Apply key testing concepts and generate effective test cases using different testing techniques.
CO5: Apply key principles of project management and quality management in software development.

Textbooks:

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

Reference Books:

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

Web Links:

1. Software Engineering Resources: https://github.com/Alliedium/awesome-softwareEngineering
2. All Software Engineering Resources: http://www.cmcrossroads.com/bradapp/links/swelinks.html
3. Resources for Software Engineering: https://www.1000sourcecodes.com/2012/05/software-engineeringresources.html#google_vignette

WEB TECHNOLOGIES			
Semester	**I**	CIE Marks	**50**
Course Code	**25MCPC507**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Provide knowledge on some basic tags of HTML.
2. Impart skills to execute simple web pages using HTML and CSS.
3. Familiarize the concepts to design layouts using advanced CSS techniques.
4. Impart knowledge to develop dynamic documents using JavaScript with CSS.
5. Disseminate knowledge to create dynamic web pages using PHP, MySQL and connection to a server.

Module 1: Introduction to HTML | **No. of Hours: 06+04**

HTML Syntax, Semantic Markup, Structure of HTML Documents: DOCTYPE, Head and Body, HTML Elements: Headings, Paragraphs and Divisions, Links, URL Relative Referencing, Inline Text Elements, Images, Character Entities, Lists, HTML5 Semantic Structure Elements: Header and Footer, Heading Groups, Navigation, Articles and Sections. Introducing Tables, Forms, Form Control elements, Table and Form accessibility.

Textbook 1 – Ch 2, 4

Laboratory Components

1. Create a simple HTML webpage with headings, paragraphs, an image, and a link.
2. Design an HTML document that includes a table and a form with various input elements.
3. Create a structured HTML5 document using semantic elements like header, footer, nav, article, and section.

Module 2: Introduction to CSS | **No. of Hours: 05+04**

Define CSS. Benefits of CSS, CSS Syntax: Selectors, Properties, Values, Location of Styles: Inline Styles, Embedded Style Sheet, External Style Sheet. Selectors: Element Selectors, Class Selectors, Id Selectors, Attribute Selectors, Pseudo-Element and Pseudo-Class Selectors, Contextual Selectors. The Cascade: How Styles Interact, The Box Model, CSS Text Styling.

Textbook 1 – Ch 3

Laboratory Components

4. Apply basic CSS styles using inline styles, embedded styles, and an external stylesheet.
5. Design a responsive webpage using media queries and a CSS framework.
6. Style an HTML form with CSS to make it visually appealing and accessible

Module 3: Advanced CSS: Layout | **No. of Hours: 05+04**

Normal Flow, Positioning Elements: Relative Positioning, Absolute Positioning. Floating Elements: Floating within a Container, Floating Multiple Items Side by Side, Containing Floats, Overlaying and Hiding Elements. Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.

Textbook 1 – Ch 5

Laboratory Components

7. Create an HTML document with elements positioned using relative and absolute positioning.
8. Design a multi-column layout using CSS floats, including a header, columns, and a footer.
9. Build a complex, responsive multi-column layout using CSS Grid.

Module 4: JavaScript: Client-Side Scripting	**No. of Hours: 05+06**

JavaScript: Client-Side Scripting, JavaScript's History and Uses. JavaScript Design Principles, Inline JavaScript, Embedded JavaScript, External JavaScript, Advanced Inclusion of JavaScript. Syntax: Variables, Comparison Operators, Logical Operators, Conditionals, Loops, Functions, Errors Using Try and Catch. JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms: Validating Forms, Submitting Forms.

Textbook 1 – Ch 6

Laboratory Components

10. Write a JavaScript script to perform basic arithmetic operations and display the results.
11. Implement form validation using JavaScript to check input fields and display error messages.
12. Use JavaScript to manipulate the DOM by changing content and styles of elements on user interaction.

Module 5: PHP with MySQL	**No. of Hours: 05+06**

PHP: PHP Tags, PHP Comments, Variables, Data Types, and Constants, Writing to Output. Program Control, Functions, Arrays, $_GET and $_POST Superglobal Arrays, $_SERVER Array, $_FILES Array. Databases and Web Development, SQL, Database APIs, Managing a MySQL Database, Accessing MySQL in PHP.

Textbook 1 – Ch 8, 11

Laboratory Components

1. Write a basic PHP script to display a message and incorporate HTML within the PHP script.
2. Handle form submissions using PHP to process and display the submitted data.
3. Connect to a MySQL database with PHP and perform CRUD operations (Create, Read, Update, Delete).

Course Outcomes:

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

CO1: Utilize foundational HTML tags in web pages.

CO2: Build static web pages with a combination of HTML for structure and CSS for styling.

CO3: Identify complex and responsive layouts that adapt to different screen sizes.

CO4: Make use of Document Object Model (DOM) in JavaScript to add, remove, or modify the content of your web pages.

CO5: Build dynamic web pages that interact with databases using PHP, a server-side scripting language.

Textbooks:

1. Randy Connolly, Ricardo Hoar, *Fundamentals of Web Development*, Pearson.

Reference Books:

1. *HTML5 Black Book*, 3rd Edition, Dreamtech Press, 2019.
2. M. Deitel, P.J. Deitel, A. B. Goldberg, *Internet & World Wide Web: How to Program*, 5th Edition, Pearson Education, 2008.

Web Links:

1. Introduction to Web Technologies and HTML 5: https://www.youtube.com/watch?v=r4F5tY_6u70
2. CSS full course for beginners: https://www.youtube.com/watch?v=n4R2E7O-Ngo
3. PHP & MySQL for beginners: https://www.youtube.com/watch?v=cGwSm8xDSwI

PROFESSIONAL COMMUNICATION AND ETHICS			
Semester	**I**	CIE Marks	**100**
Course Code	**25MCNM508**	SEE Marks	**-**
Teaching Hours/Week (L:T:P)	**2:0:0**	Exam Hours	**-**
Total Hours	**26**	Credits	**-**

Course Learning Objectives:

This course is designed to

1. Impart knowledge on the fundamental concepts of interpersonal and professional communication.
2. Provide knowledge on active listening with focus on content, purpose, ideas and tone.
3. Facilitate fluent speaking skills in social, academic and professional situations.
4. Disseminate effective writing skills in business, technology and academic arenas.
5. Instil essential social and professional skills necessary to navigate various interpersonal interactions effectively

Module 1: Fundamentals of Communication	**No. of Hours: 05**

Introduction, An Overview of Communication, Process of Communication, Purpose of Professional Communication, Different Forms of Communication, Communication Network in an Organization, Barriers to Communication, Some Remedies.

Textbook 1 – Ch 1

Module 2: Listening Skills	**No. of Hours: 05**

Developing Effective Listening Skills: Listening Is an Art, Listening vs Hearing, Poor Listening vs Effective Listening, Advantages of Good Listening, Process of Listening, Types of Listening, Intensive Listening vs Extensive Listening, Barriers to Effective Listening, Five Steps to Active Listening, Techniques for Effective Listening.

Textbook 1 – Ch 3

Module 3: Speaking Skills	**No. of Hours: 05**

Non-verbal Communication: Body Language, Paralinguistic Features, Proxemics/Space Distance.
Group Discussions: Introduction, Difference Between GD and Debate, Personality Traits to be Evaluated, Dynamics of Group Behaviour/Group Etiquette and Mannerisms.
Job Interviews: Introduction, Process, Stages in Job Interviews, Types, Desirable Qualities, Preparation, Using Proper Verbal and Non-verbal Cues, Exhibiting Confidence.

Textbook 1 – Ch 4

Module 4: Writing Skills	**No. of Hours: 05**

Business Letters, Elements of Style, Types of Business Letters: Acknowledgement Letter, Letter of Recommendation, Appreciation Letter, Acceptance Letter, Apology Letter, Complaint Letter, Adjustment Letter, Inquiry Letter, Permission Letter, Invitation Letter, Rejection Letter, Order Letter, Application Letters and Cover Letters, Sales Letters.
Resume Preparation, Business Reports: Structure of Formal Reports, Email and Blog Writing.

Textbook 1– Ch 6

Module 5: Effective Presentations and Etiquettes	**No. of Hours:06**

Presentation: Essential Characteristics of a Good Presentation, preparing a presentation, delivering the presentation,
Business Etiquettes: Introductions, Formal Dress Code, Formal Dining Etiquettes, Conflict

management through communication.

Textbook 2 – Ch 14,15,16

Course Outcomes:

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

CO1: Show how to eliminate barriers and use verbal/non-verbal cues at social and workplace situations.
CO2: Outline listening strategies to comprehend wide-ranging vocabulary, grammatical structures, tone and pronunciation.
CO3: Interpret effective speaking skills at social, academic and business situations.
CO4: Interpret effective writing skills for drafting academic, business and technical documents.
CO5: Outline the necessary skills and knowledge to navigate social and professional interactions with confidence and grace.

Textbooks:

1. Sanjay Kumar & Pushp Lata, "*Communication Skills with CD",* New Delhi: Oxford University Press, 2018
2. P. D. Chaturvedi, Mukesh Chaturvedi, *"Business Communication",* Pearson, 2012

Reference Books:

1. Hemphill, P.D., McCormick, D. W., & Hemphill, R. D. "*Business Communication with writing improvement exercises",* Upper Saddle River, NJ: Prentice Hall, 2001
2. Raman, M., & Sharma S, "*Technical Communication: Principles and practice",* New Delhi, Oxford University Press, 2016
3. Kaul, A. *Effective Business Communication,* Prentice Hall India Learning Pvt Ltd, 2015.
4. Rizvi, A. M. *Effective Technical Communication: A guide for Scientists and Engineers,* New Delhi, Tata McGraw Hill, 2010

Web Links:

1. Professional Communication Skills:https://www.onestopenglish.com/adults/business-and-esp/esp- lesson-plans/professional-communication-skills
2. Effective Communication: https://communicationprogram.wharton.upenn.edu/lib ary/
3. Professional Communication: https://www.futurelearn.com/courses/effective-communication- skills-for-professionals
4. Business Etiquette: https://www.youtube.com/watch?v=15XC6R80wb4

II SEM

DESIGN AND ANALYSIS OF ALGORITHMS			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPC511**	SEE Marks	**50**
Teaching Hours /Week (L: T:P)	**4:0:0**	Exam Hours	**03**
Total Hours	**52**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Provide a solid foundation in the basic concepts of algorithms, including their types, structure, and efficiency in solving computational problems.
2. Instil the ability to analyse the asymptotic performance of algorithms and evaluate their efficiency.
3. Familiarize students with various algorithm design paradigms, such as Divide and Conquer, Greedy Method, and Dynamic Programming, through illustrative problem-solving.
4. Instil the knowledge required to design, optimize, and implement efficient algorithms tailored to real- world engineering scenarios, incorporating advanced techniques like Backtracking, Branch and Bound, and Randomized Algorithms.

Module 1: Introduction to Algorithm Analysis	**No. of Hours: 12**

Introduction to Algorithms: Need for Algorithm Efficiency, Fundamental Stages of Problem Solving, Basics of Algorithm Analysis: Basics of Algorithm Complexity, Introduction to Time complexity, Analysis of Iterative Algorithms, Rate of Growth, Asymptotic Analysis, Space Complexity Analysis. Important Problem Types: Sorting, Searching.

Textbook 1 - Ch 1, 2
Textbook 2 - Ch 1

Module 2: Divide and Conquer, Decrease and Conquer	**No. of Hours: 10**

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. Advantages and Disadvantages of decrease and conquer.

Textbook 1 - Ch 4, 5, 6
Textbook 2 - Ch 3

Module 3: Greedy Approaches	**No. of Hours: 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.

Textbook 1 - Ch 9
Textbook 2 - Ch 4

Module 4: Dynamic Programming	**No. of Hours: 10**

General method with Examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Bellman- Ford Algorithm, Travelling Salesman problem.

Textbook 1 - Ch 8
Textbook 2 - Ch 5

Module 5: Backtracking	**No. of Hours: 10**

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.

Textbook 1 - Ch 12
Textbook 2 - Ch 8, 11

Course Outcomes:

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

CO1: Apply the fundamental concepts of algorithms, including asymptotic analysis and performance evaluation, to analyse space and time complexities for various algorithms.

CO2: Demonstrate the ability to solve problems using Divide and Conquer, Decrease and Conquer, and Transform and Conquer approaches, and analyse their efficiency in real-world scenarios.

CO3: Apply Greedy algorithms to solve optimization problems such as Knapsack, Job Sequencing, and Minimum Spanning Tree, and evaluate their performance.

CO4: Apply dynamic programming techniques to solve optimization problems using algorithms like Warshall's, Floyd's, Bellman-Ford, Knapsack, and TSP.

CO5: Implement Backtracking and Branch & Bound methods to solve combinatorial and optimization problems such as N-Queens, Graph Colouring, and Travelling Salesperson.

Textbooks:

1. Anany Levitin, "*Introduction to the Design and Analysis of Algorithms*", 3rd Edition, Pearson Education, 2012
2. Ellis Horowitz, Sartaj Sahni and Rajasekaran, "*Computer Algorithms/C++*", 2nd Edition, Universities Press, 2014

Reference Books:

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

Web Links:

1. Design and Analysis of Algorithms: https://www.codechef.com/learn/course/college-design-analysis- algorithms

2. Algorithms: Design and Analysis: https://online.stanford.edu/courses/soe-ycsalgorithms1-algorithms-design-and-analysis-part-1
3. Design and Analysis of Algorithms: https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/

PROGRAMMING IN JAVA			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPC512**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hours	**03**
Total Hours	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Impart the fundamental concepts of Object-Oriented Programming.
2. Classify the different programming constructs used in Java Programming.
3. Use the concepts of multiple inheritance and interface in solving real-time problems.
4. Enable students to use exception handling techniques while solving programming problems.
5. Build applications using the concepts of multithreading.

Module 1: OOPS Concepts and Java Programming | **No. of Hours: 08+04**

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.

Textbook 1- Ch 1, 2, 3

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: Introduction to Java | **No. of Hours: 08+04**

Basics of Java: Features of Java, Byte Code and Java Virtual Machine, JDK, Data types, Operator, Control Statements – If, else, nested if, if-else ladders, Switch, while, do-while, for, for-each, break, continue.

Array and String: Single and Multidimensional Array, String class, StringBuffer class, Operations on string, Command line argument, Use of Wrapper Class.

Textbook 1 - Ch 4, 5, 7

Laboratory Component

3. Write a Java program that calculates and displays grades based on student scores stored in an array.
4. Write a Java program that analyzes a given text string and provides statistics on characters, words, and sentences.

Module 3: Multiple inheritance and interface	**No. of Hours:08+06**

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.
Textbook 1 - Ch 8

Laboratory Components

5. 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.
6. Write a Java program to demonstrate Multiple inheritance using interfaces and to calculate the area of a rectangle and triangle.

Module 4: Exception Handling	**No. of Hours:08+04**

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.
Textbook 1 – Ch 10

Laboratory Components

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

Module 5: Multi-Threaded programming	**No. of Hours:08+06**

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.
Textbook 1 – Ch 12

Laboratory Components

9. Write a Java program to create multiple threads using different thread methods.
10. Write a Java program to perform the following operations.
    a) Thread 1: Display the natural numbers from 1 to 100.
    b) Thread 2: Display even numbers from 2 to 100
    c) Thread 3: Display odd numbers from 1 to 99.
    Set Thread Properties so that thread 2 must run at beginning followed by thread 3 and at last thread 1 must be executed.

Course Outcomes:

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

CO1: Apply OOP principles and Java programming concepts to develop functional programs solving real- world problems.
CO2: Develop Java programs using control statements, arrays, and string handling.
CO3: Develop Java programs using inheritance, polymorphism, and interfaces for modular code.
CO4: Construct Java programs that handle exceptions using try-catch, throw, and custom exception classes.
CO5: Apply Java multithreading concepts, including synchronization and thread communication, to develop programs.

Textbooks:

1. Herbert Schildt and Dale Skrien, "*Java Fundamentals – A Comprehensive Introduction"*, 1st Edition, McGraw Hill, 2013
2. Herbert Schildt, "*Java the Complete Reference",* 7th Edition, McGraw Hill, 2011
3. T. Budd, "*Understanding Object-Oriented Programming with Java",* Pearson Education, 1999

Reference Books:

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

Web Links:

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

DATABASE MANAGEMENT SYSTEMS			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPC513**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hours	**03**
Total Hours	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Provide a strong foundation in database concepts, technology, and practice.
2. Develop SQL programming skills through a variety of database problems.
3. Impart knowledge on the use of concurrency and transactions in a database system.
4. Deliver database applications for real-world problems.

Module 1: Introduction to Databases	**No. of Hours: 08+06**

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.

Textbook 1 – Ch 1, 2, 3

Laboratory Components:

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 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 are more than 2.
e. Find out the Publisher who has published more than 2 books.

Module 2: Relational Model	**No. of Hours: 08+06**

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.
Textbook 1 – Ch 5, 8, 9

Laboratory Components:

3. 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, many teams are contesting each having a Teamid, Team_Name, City and 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 is played between the two teams on 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's name where all its won matches played in the same stadium.

Module 3: Structured Query Language	**No. of Hours:08+06**

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.
Textbook 1 – Ch 6, 7

Laboratory Components:

4. A country wants to conduct an election for parliament. A country has 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. Many candidates are 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 for 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 belongs 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 Hours: 08+06**

Normalization: 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.

Textbook 1 – Ch 14

Laboratory Components:

5. 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 places visited by the tourists of all countries.

Module 5: Transaction Processing and Concurrency Control	**No. of Hours:08**

Transaction Processing: Introduction to Transaction Processing, Desirable properties of Transactions, Characterizing schedules based on Recoverability and 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.

Textbook 1 – Ch 20, 21

Course Outcomes:

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

CO1: Apply ER modeling concepts to design databases and map them to relational schemas in a DBMS.

CO2: Utilize the concepts of relational model and relational algebra to solve a given problem definition.

CO3: Develop SQL queries for database manipulation.

CO4: Apply the concepts of normalization and design effective databases.

CO5: Use simple database systems to relate the concept of transaction, concurrency control and recovery.

Text Books:

1. Ramez Elmasri, Shamkant B. Navathe, "*Fundamentals of Database System*", 7th Edition, Pearson, 2017

Reference Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "*Database System Concepts*", 6th Edition, Tata McGraw Hill Education Private Limited, 2011
2. Ramakrishnan and Gehrke, "*Database Management Systems*", 3rd Edition, McGraw Hill, 2014

Web Links:

1. Database Management System: https://onlinecourses.nptel.ac.in/noc22_cs91/preview
2. Introduction to Databases: https://archive.nptel.ac.in/courses/106/106/106106220/
3. Database Management Essentials: https://www.coursera.org/learn/database-management

OBJECT ORIENTED MODELING AND DESIGN			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPC514**	SEE Marks	**50**
Teaching Hours/Week (L: T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to:

1. Impart core Object-Oriented Principles while architecting and designing software applications.
2. Provide students with a comprehensive understanding of how OOMD fits into the broader context of the software development lifecycle.
3. Equip students with the tools and techniques to model and design software using object-oriented methodologies.
4. Critically assess object-oriented models for correctness, completeness, and maintainability.
5. Perform design reviews and walk-throughs to identify potential improvements in the system architecture.
6. Recognize the benefits and challenges of object-oriented modeling in large-scale system development.

Module 1: Introduction to Object-Oriented Modeling	**No. of Hours:06+06**

Object Orientation in Object-Oriented development, Usefulness of Object-Oriented Development, Object-Oriented modeling history, Modeling Concepts: Modeling as a design technique, Class modeling : Link and Association concepts, Generalization and Inheritance, A sample class model, Navigation of class models, Advanced Class Modeling: Association Ends, N-ary Associations, Aggregation, History and evolution of UML, UML Diagrams: Overview of Use Case, Class, Object, Sequence, Collaboration, Activity, Component, Deployment Diagrams, Tools for UML modeling.

Textbook 1 - Ch 1, 2, 3, 4, 7, 15

Textbook 2 - Ch 5, 7, 9, 10

Lab Component:

1. Model a library management system by identifying and defining classes, their attributes and methods. Create associations, generalizations and dependencies between classes.
2. Develop a use-case diagram for an online shopping application. Identify the actors, use cases and establishing relationships among the use cases.

Module 2: Object-Oriented Modeling – State Modeling and Advanced State Modeling	**No. of Hours:05+06**

State Modeling: Events, States, Transition and conditions, State diagrams, State diagram behavior, Advanced State Modeling: Nested state diagrams, Nested states, Signal Generalization, Concurrency, A sample state model, Relation of class and state models.

Textbook 1 - Ch 5, 6

Lab Component:

3. Create a state diagram for an “order” in an ecommerce system. Define the different states (“Pending”, “Processed” and “Shipped” and the events that cause these transitions.
4. Use the UML tool to represent the different states while booking an airline ticket. Show the events that trigger the transitions.

Module 3: Object Oriented Modeling – Interaction Modeling and Advanced Interaction Modeling	**No. of Hours: 05+06**

Interaction Modeling: Use Case models, Sequence models, Activity models, Advanced Interaction Modeling: Use Case relationships, Procedural sequence models, Special constructs for activity models, Concepts summary: Class model, State model, Interaction model, Relationship among the models.
Textbook 1 - Ch 7, 8

Lab Component:

5. Model a sequence diagram for placing an order in an e-commerce platform. Identify the objects involved and the sequence of messages exchanged between them to complete the process.
6. Create an activity diagram to model a payment processing workflow in an online shopping application. Define the activities, decision points and parallel activities to visualize the flow.

Module 4: Object Oriented Design Principles	**No. of Hours: 05+06**

SOLID Principles: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion, Design by Contract: Preconditions, Postconditions, Invariants, Implementation of Object-Oriented Designs: Translating UML diagrams into code, Refactoring, Unit Testing and Test- Driven Development.
Textbook 4 - Ch 4, 5, 8, 9, 10, 11, 12

Lab Component:

7. Create a collaboration diagram to model the interactions involved in processing a customer's order.
8. Model the deployment of an e-commerce application across a distributed network.

Module 5: Object-Oriented Design Patterns	**No. of Hours: 05**

Design Patterns: What is a design pattern, Describing design patterns, The catalog of design patterns, Organizing the catalog, How design patterns solve design problems, Selecting design patterns, Using design patterns, Creational patterns: prototype and singleton, Structural patterns: adaptor and proxy.
Textbook 2 - Ch 1, 3, 4

Course Outcomes:

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

CO1: Identify the system requirements for a given problem, key objects and relationships and model applications effectively in an Object-Oriented framework.
CO2: Model the state of an object or a system for different use-case scenarios through state diagrams, and collaboration diagrams illustrating transitions and actions.
CO3: Model object interactions and the flow of control during specific use cases or system scenarios.
CO4: Develop the ability to incorporate SOLID principles in Object-Oriented design to create modular, scalable and maintainable systems.
CO5: Apply relevant design patterns to solve common design problems and improve the overall software structure.

Software Tools: UML Modeling Tools (StarUML/ArgoUML/BOUML)

Textbooks:

1. Michael Blaha and James Rumbaugh, "*Object-Oriented Modeling and Design with UML",* 2nd Edition, Pearson Education India, 2005
2. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides: "*Design Patterns –Elements of Reusable Object-Oriented Software"*, Pearson Education, 2007
3. Martin Fowler, "*UML Distilled: A Brief Guide to the Standard Object Modeling Language"*, 3rd Edition, Addison-Wesley, 2003
4. Martin C. Robert, Martin Micah, "*Agile Principles, Patterns and Practices in C#,* Prentice Hall, 2006

Reference Books:

1. Grady Booch, "Object-Oriented Analysis and Design with Applications", 3rd Edition, Addison-Wesley, 2007
2. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", 3rd Edition, Addison-Wesley Professional, 2004
3. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design", 2nd Edition, Prentice Hall Professional, 2002
4. Brett McLaughlin, Gary Pollice, and David West, "Head First Object-Oriented Analysis and Design", 1st Edition, O'Reilly Media, Inc., 2006
5. Grady Booch, James Rumbaug, and Ivar Jacobson, "The Unified Modeling Language User Guide", 2nd Edition, Addison-Wesley, 2005

Web Links:

1. UML Diagrams - Introduction to Object-Oriented Modeling on Visual Paradigm: https://online.visual-paradigm.com/diagrams/features/uml-tool/
2. SOLID Principles in Object-Oriented Design: https://stackify.com/solid-design-principles/
3. Articles on Object Oriented Modeling and Design: https://www.getallarticles.com/category/object-oriented-modeling-and-design/

RESEARCH METHODOLOGY AND IPR			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCHM515**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Provide knowledge on how to define and approach a research problem.
2. Make students familiar with how to formulate a literature survey.
3. Instill skills to develop theoretical and conceptual frameworks and write an effective review.
4. Impart knowledge to perform data collection and know the importance in arriving at solutions.
5. Familiarize with the interface of IPR in today's real time scenario.

Module 1: Introduction to Research Methodology	**No. of Hours: 09**

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, Research Process.

Textbook 1 – Ch 1

Module 2: Defining the Research Problem	**No. of Hours: 09**

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, 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.

Textbook 1 – Ch 2

Textbook 2 – Ch 3

Module 3: Research Design	**No. of Hours: 08**

Research Design: Meaning of Research Design, Need for Research Design, Basic Principles of Experimental Designs.

Sample Design: Characteristics of a Good Sample Design, Different Types of Sample Designs, Selecting

a Random Sample.

Textbook 1 – Ch 3, 4

Module 4: Data Collection	**No. of Hours: 08**

Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, Collection of Secondary Data. Selection of Appropriate Method for Data Collection.

Report Writing: Different Steps in Writing Report, Mechanics of Writing a Research Report,

Precautions for Writing Research Reports.

Textbook 1 – Ch 6, 14

Module 5: Intellectual Property (IP) Acts	**No. of Hours: 08**

Intellectual Property (IP) Acts: Introduction to IP: Introduction to Intellectual Property (IP), different types of IPs and its importance in the present scenario. Infringement Act on Patent, Cyber Law, Copyright, Trade Mark.

Textbook 3 – Ch 3, 4

Course Outcomes:

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

CO1: Apply various research methods and articulate the phases of the research process on different problems and arrive at an acceptable solution.

CO2: Make use of appropriate functions of the literature review in research, carrying out a literature search, developing theoretical and conceptual frameworks.

CO3: Identify the procedure to collect and categorize data as primary and secondary, interpret its meaning and write effective reports.

CO4: Develop a research design and identify appropriate data sampling techniques to solve a research problem.

CO5: Identify infringements on intellectual property and categorize them as CopyRight Act/Patent Act/Cyber Law/Trademark.

Textbooks:

1. C.R. Kothari, *Research Methodology, Methods and Techniques,* 4th edition, Gaurav Garg New Age International, 2018
2. Ranjit Kumar, *Research Methodology a step-by- step guide for beginners,* 3rd Edition, SAGE Publications, 2011.
3. Dr. Santosh M Nejakar, Dr. Harish Bendigeri Research Methodology and Intellectual Property Rights, 4th Edition, 2023.

Reference Books:

1. David V. Thiel "Research Methods for Engineers" Cambridge University Press.
2. N.K.Acharya Intellectual Property Rights, 6th Edition Asia Law House, 2013.

Web Links:

1. Introduction to Research Methodology: https://www.enago.com/academy/choose-best-research- methodology/
2. Research Methodologies: https://library.tiffin.edu/researchmethodologies/whatareresearchmethodologies
3. IPR and Patents: https://www.editage.com/insights/patents-101-what-are-patents-and-how-do- you-acquire-them?refer=scroll-to-2-article&refer-type=article.

CRYPTOGRAPHY AND CYBERSECURITY			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPE521**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Impart Knowledge on basic concepts of Computer security and Cryptography
2. Provide skills to utilize Symmetric and Asymmetric cryptographic algorithms, Message Authentication algorithms and Hash functions.
3. Familiarize various types of cyber-attacks and cyber-crimes and an overview of the cyber laws.
4. Instill the ability to analyse cyber security challenges, assess their implications and identify the nature of cybercrime in real-world scenarios.

Module 1: Attacks on Computers, Computer Security and Cryptography	**No. of Hours: 10**

Attacks on Computers, Computer Security: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security mechanisms, A model for Network Security

Cryptography: Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography.

Textbook 1 - Ch 1, 3

Module 2: Symmetric Ciphers	**No. of Hours: 08**

Symmetric key Ciphers: Block Cipher principles & Algorithms (DES, AES), Block cipher modes of operation, Stream ciphers.

Asymmetric key Ciphers: Principles of public key cryptosystems, Algorithms (RSA, Diffie Hellman).

Message Authentication Algorithms and Hash Functions: Authentication requirements, Functions, Message authentication codes, Hash Functions, Secure hash algorithm.

Textbook 1 - Ch 4, 6, 9, 10,12

Module 3: Cyber Security	**No. of Hours: 08**

Classifications of Cyber Crimes, Tools and Methods: Password Cracking, Keyloggers, Spywares, SQL Injection, Buffer Overflow, Attacks on Wireless Networks.

Textbook 2 - Ch 1, 4

Module 4: Cyber Security and privacy	**No. of Hours: 08**

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing, and the associated challenges for organizations.

Textbook 2 - Ch 9

Module 5: Cybercrime and Cyber terrorism	**No. of Hours: 08**

Cybercrime and Cyber terrorism: Introduction, intellectual property in cyberspace, the ethical dimension of cybercrimes, the psychology, mindset, and skills of hackers and other cyber criminals. Cybercrime: Examples and Mini-Cases Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances. Mini-Cases: The Indian Case of Online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

Textbook 2 - Ch 10, 11

Course Outcomes:

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

CO1: Apply fundamental principles of computer security and cryptography to identify and analyse different types of security attacks and mechanisms.

CO2: Utilize symmetric and asymmetric cryptographic techniques, including DES, AES, RSA, and Diffie-Hellman, to design secure communication systems.

CO3: Demonstrate the ability to apply tools and methods, such as password cracking, keyloggers, and SQL injection, to assess and mitigate cyber threats.

CO4: Analyze the organizational implications of cyber security, including the impact of web threats, privacy issues, and social media security risks, to propose appropriate mitigation strategies.

CO5: Investigate real-world cases of cybercrime and cyber terrorism, evaluating their psychological, ethical, and societal dimensions to derive actionable insights

Textbooks:

1. William Stallings, *"Cryptography and Network Security"*, 8th Edition, Pearson Education, 2020
2. Nina Godbole and Sunit Belpure, "*Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives",* Wiley, 2011

Reference Books:

1. C K Shyamala, N Harini, Dr T R Padmanabhan, *"Cryptography and Network Security"*, Wiley India, 2011
2. Forouzan Mukhopadhyay, "*Cryptography and Network Security"*, 3rd Edition, McGraw Hill, 2015
3. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, "*Computer and Cyber Security: Principles, Algorithm, Applications and Perspectives"*, CRC Press Inc, 2019

Web Links:

1. Using Cryptography Correctly: https://online.stanford.edu/courses/xacs130-using-cyptography- correctly.
2. Introduction to Cryptography: https://www.mygreatlearning.com/academy/learn-for- free/courses/introduction-to-cryptography
3. Introduction to Cybersecurity(netcad): https://www.netacad.com/courses/introduction- to-cybersecurity?courseLang=en-US

NETWORK SECURITY			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPE522**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Gain a solid foundation in network security principles.
2. Understand cryptographic techniques to secure communication.
3. Analyze various network threats, vulnerabilities, and mitigation strategies.
4. Configure and manage essential network security tools and protocols.

Module 1: Network Security Fundamentals	**No. of Hours: 08**

Introduction to Network Security Concepts: Need for Security, Security Attacks (Passive, Active), Security Services (Confidentiality, Integrity, Availability), Security Mechanisms (Encryption, Authentication, Access Control), Security Models (CIA Triad, Bell-LaPadula Model)
Symmetric Key Cryptography: Classical Encryption Techniques (Caesar Cipher, Vigenere Cipher), Modern Block Ciphers (DES, AES), Operating Modes (CBC, ECB, CTR).
Textbook 1: Ch 1, 3, 6

Module 2: Cryptographic Techniques	**No. of Hours: 08**

Advanced Encryption Standard: Finite Field Arithmetic, AES Structure, AES Transformation Functions, AES Key Expansion, An AES Example, AES Implementation.
Principles of Public-Key Cryptosystems, Public-Key Cryptosystems, Applications for Public-Key Cryptosystems, Requirements for Public-Key Cryptography, Public-Key Cryptanalysis, The RSA Algorithm, The Security of RSA
Textbook 1: Ch 7, 9

Module 3: Network Security Applications	**No. of Hours: 08**

Network Access Control and Cloud Security, Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, Cloud Security Risks and Countermeasures, Data Protection in the Cloud, Addressing Cloud Computing Security Concerns, Transport-Level Security, Web Security Considerations, Transport Layer Security, HTTPS, Secure Shell (SSH)
Textbook 1: Ch 16, 17

Module 4: Wireless Network Security	**No. of Hours: 09**

Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security. User Authentication, Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User-Authentication Using Asymmetric Encryption, Federated Identity Management, Personal Identity Verification.
Textbook 1: Ch 15, 18

Module 5: Email & IP Security Management	**No. of Hours: 09**

Electronic Mail Security, Internet Mail Architecture, Email Formats, Email Threats and Comprehensive Email Security, S/MIME, Pretty Good Privacy, DNSSEC, DNS-Based Authentication of Named Entities, Domain-Based Message Authentication, Reporting, and Conformance IP Security, IP Security,

IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange, Cryptographic Suites.
Textbook 1: Ch 19, 20

Course Outcomes:

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

CO1: Apply security principles to identify and categorize real-world network threats.
CO2: Apply cryptographic algorithms and key management techniques to secure communication channels.
CO3: Apply secure communication protocols (TLS/SSL, VPNs) to configure secure connections.
CO4: Apply intrusion detection/prevention systems (IDS/IPS) and firewall configurations to mitigate network security threats.
CO5: Utilize network security monitoring tools (Wireshark, SIEM) to analyze network traffic and identify security incidents.

Textbooks:

1. William Stallings, *"Cryptography and Network Security: Principles and Practice"*, 7th Edition, Pearson Education Limited, 2017

Reference Books:

1. William Stallings and Michael Brown, *"Network Security Essentials: Applications and Standards"*, 4th Edition, Pearson Education Limited, 2011
2. Matt Bishop, *"Computer Security: Art and Science"*, 2nd Edition, Pearson Education Limited, 2009

Web Links:

1. Public key encryption: https://www.khanacademy.org/computing/computers-and- internet/xcae6f4a7ff015e7d:online-data-security/xcae6f4a7ff015e7d:data-encryption- techniques/a/public-key-encryption
2. Transport Layer Security: https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security
3. 10 Risk Factors: https://securityscorecard.com/blog/securityscorecard-10-risk-factors-explained/
4. SIEM Guide: https://www.elastic.co/guide/en/siem/guide/index.html

MACHINE LEARNING			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPE523**	SEE Marks	**50**
Teaching Hours./Week (L: T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Familiarize with the core concepts of Machine Learning.
2. Impart knowledge to apply common supervised learning algorithms to solve classification and regression problems.
3. Provide knowledge on techniques essential to uncover hidden patterns and structures within unlabeled data.
4. Familiarize with Multilayer Perceptron via back propagation algorithm to solve classification problems.
5. Provide knowledge on Reinforcement learning techniques.

Module 1: Introduction to Machine Learning	**No. of Hours: 06+04**

Introduction, **Examples of Machine Learning Applications**: Learning Associations, Classification, Regression, Unsupervised Learning, Reinforcement Learning, Data Preparation Tasks: Data Cleaning, Feature Selection, Data Transform feature engineering, Dimensionality Reduction, **Data Preparation**: Problems with Naive Data Preparation, Train and Test Dataset, K-Fold Cross validation, Data Cleaning: Basics of Data Cleaning, Outlier Identification and Removal, Marking and Remove Missing Data, Statistical Imputation.

Textbook 1 - Ch 1
Textbook 3 - Ch 3, 4

Laboratory Components:

1. Demonstrate the following concepts of data preprocessing techniques by using few datasets.
   a. Identify and remove outliers using IQR and Local Outlier Factor methods.
   b. Identify and remove missing values.
   c. Selection of numerical and Categorical features.
   d. Perform Data Transformation.

Module 2: Supervised Learning	**No. of Hours: 05+04**

Learning Multiple Classes, Regression, Bayesian Decision Theory: Introduction, Classification, **Parametric Methods:** Maximum Likelihood Estimation, Bernoulli Density, Multinomial Density, Gaussian (Normal) Density, Regression.

Textbook 1 - Ch 2, 3, 4

Laboratory Components:

2. Linear Regression (Prediction): Predict a continuous value based on a linear relationship with features.
3. Logistic Regression (Classification): Classify data points into two categories (0 or 1) based on features.

Module 3: Unsupervised Learning	**No. of Hours: 05+06**

Clustering: Introduction, Mixture Densities, k-Means Clustering, Hierarchical Clustering, **Nonparametric Methods:** Nonparametric Density Estimation, Histogram Estimator, Kernel Estimator, k-Nearest Neighbor Estimator, Decision Trees: Introduction, Univariate Trees, Classification Trees.
Textbook 1 - Ch 7,8,9

Laboratory Components:

4. K-Nearest Neighbors (KNN) – Classification: Classify data points based on the majority vote of their K nearest neighbors.
5. Decision Tree - Classification: Classify data points based on a tree-like structure with decision rules.

Module 4: Multilayer Perceptrons	**No. of Hours:05+04**

Introduction, Understanding the Brain, Neural Networks as a Paradigm for Parallel Processing, The Perceptron, Training a Perceptron, Multilayer Perceptron, Backpropagation Algorithm, Nonlinear Regression, Two-Class Discrimination, Multiclass Discrimination.
Textbook 1 - Ch 11

Laboratory Components:

6. Perceptron: Implement a perceptron and train it to perform classification for a given problem.

Module 5: Reinforcement Learning	**No. of Hours: 05+06**

Elements of Reinforcement Learning, Model-Based Learning, Value Iteration, Temporal Difference Learning, Exploration Strategies, Deterministic Rewards and Actions, Nondeterministic Rewards and Actions, Eligibility Traces, Generalization.
Textbook 1 - Ch 16

Laboratory Components:

7. Develop a reinforcement learning agent to control a car on a hill to reach the flag at the top.

Course Outcomes:

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

CO1: Make use of appropriate machine learning paradigm for a given problem.
CO2: Utilize supervised learning models to make predictions based on labeled data.
CO3: Identify meaningful insights from unlabeled data through clustering.
CO4: Apply Multilayer Perceptron to solve classification problems.
CO5: Outline the different reinforcement learning techniques.

Textbooks:

1. Ethem Alpaydin, "*Introduction to Machine Learning*", 4th Edition, MIT Press, Prentice Hall of India, 2020
2. Andreas C. Müller and Sarah Guido, "*Introduction to Machine Learning with Python A Guide for Data Scientists*", 4th Edition, O'Reilly Media, Inc, 2017
3. Jason Brownlee, "*Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*", 2nd Edition, Machine Learning Mastery, 2020

Reference Books:

1. Tom Mitchell, *"Machine Learning"*, 1st Edition, McGraw Hill, 2017
2. George F Luger, *"Artificial Intelligence: Structures and Strategies for Complex Problem Solving"*, 6th Edition, Pearson Education, 2021
3. Stephen Marsland, *"Machine Learning - An Algorithmic Perspective"*, 2nd Edition, CRC press, 2018

Web Links:

1. Machine Learning Tutorial: https://www.tutorialspoint.com/machine_learning/index.htm
2. A Brief Introduction to Machine Learning: https://onlinecourses.nptel.ac.in/noc23_cs18/preview
3. Machine Learning Full Course: https://www.youtube.com/watch?v=GwIo3gDZCVQ
4. Data Preparation for Machine Learning in Python: https://www.youtube.com/watch?v=ICoiTcY2uFY

PATTERN RECOGNITION			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCPE524**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:0:2**	Exam Hours	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to:

1. Impart knowledge on the fundamental concepts of the classification process.
2. Provide knowledge on how to apply statistics in pattern recognition.
3. Demonstrate nonparametric learning methods to understand the principles behind artificial neural networks and kernel machines.
4. Instill the basic principles of feature extraction in image processing.
5. Perform dimensionality reduction using varied techniques and estimate classifier performance.

Module 1: Fundamentals of Classification	**No. of Hours: 06 + 04**

The Classification Process, Features, Training and Learning, Supervised Learning and Algorithm Selection, Approaches to Classification.

Non Metric Methods: Introduction, Decision Tree Classifier: Information, Entropy, and Impurity, Information Gain, Decision Tree Issues. Rule-Based Classifier.

Textbook 1 – Ch 2, 3

Laboratory Components

1. Implement a program to perform the following tasks:
   a. Load a dataset (e.g., Iris dataset or any other classification dataset).
   b. Implement a decision tree classifier using Scikit-Learn library.
   c. Calculate and display the information gain, entropy, and impurity for each feature in the dataset.
   d. Fit the decision tree model to the training data and visualize the tree structure using a plotting library (e.g., Matplotlib).
   e. Evaluate the model's performance on a test dataset by calculating accuracy, precision, recall, and F1-score.
2. Write a program to perform the following tasks:
   - Load a dataset (e.g., the Titanic dataset or any relevant dataset).
   - Define classification rules based on specific features (e.g., age, gender, class) and implement a simple rule-based classifier.
   - Create a function that applies these rules to classify instances in the dataset (e.g., predicting survival based on rules).
   - Test the classifier on a subset of data and report the classification results.
   - Compare the performance of the rule-based classifier with that of a decision tree classifier using the same dataset and report findings.

Module 2: Statistics in Pattern Recognition	**No. of Hours: 05 + 04**

Measured Data and Measurement Error, Probability Theory: Simple Probability Theory, Conditional Probability and Bayes' Rule, Naıve Bayes Classifier. Continuous Random Variables: The Multivariate Gaussian, The Covariance Matrix, the Mahalanobis Distance.

Textbook 1 – Ch 4

Laboratory Components

3. Write a program to perform the following tasks:
   a. Load a dataset (e.g., the Iris dataset or any relevant dataset).
   b. Preprocess the data by handling missing values and encoding categorical variables if necessary.
   c. Implement a Naïve Bayes classifier using the following steps:
      - Calculate the prior probabilities for each class.
      - Calculate the likelihood of each feature given the class.
      - Use Bayes' Rule to classify the instances in the test set.
   d. Evaluate the performance of the classifier using metrics such as accuracy, precision, recall, and F1-score.
   e. Visualize the results using a confusion matrix to illustrate the classification performance.
4. Write a program to perform the following tasks:
   a. Load a multivariate dataset (e.g., a dataset with features like height, weight, and age).
   b. Calculate the mean vector and covariance matrix of the dataset.
   c. Implement a function to compute the Mahalanobis distance for each instance in the dataset.
   d. Use the Mahalanobis distance to identify anomalies (instances that are significantly far from the mean).
   e. Visualize the data points, the mean, and the identified anomalies on a scatter plot.

Module 3: Parametric and Nonparametric Learning	**No. of Hours: 05+06**

Parametric Learning: Bayesian Decision Theory, Discriminant Functions and Decision Boundaries, MAP (Maximum A Posteriori) Estimator.
Nonparametric Learning: Histogram Estimator and Parzen Windows, k-Nearest Neighbour (k-NN) Classification, Artificial Neural Networks, Kernel Machines.
Textbook 1 – Ch 5, 6

Laboratory Components

5. Write a program to perform the following tasks:
   a. Load a dataset (e.g., a binary classification dataset like the Breast Cancer Wisconsin dataset or any relevant dataset).
   b. Pre-process the data, handling any missing values or categorical variables as needed.
   c. Implement the Bayesian decision technique for classification:
      - Calculate prior probabilities and likelihoods for each class.
      - Compute the posterior probability for each class using Bayes' Theorem.
   d. Implement the Maximum A Posteriori (MAP) estimator to classify the data points.
   e. Evaluate the performance of the classifier using metrics such as accuracy, precision, recall, and F1-score.
6. Write a program to perform the following tasks:
   a. Load a dataset (e.g., the Iris dataset or any other suitable dataset for classification).
   b. Split the dataset into training and testing sets.
   c. Implement the k-Nearest Neighbour (k-NN) classification algorithm:
      - Calculate distances between the test instance and all training instances.
      - Identify the k-nearest neighbours and perform majority voting to classify the test instance.
   d. Implement a parametric learning approach, such as logistic regression, to classify the same

dataset.

e. Compare the performance of the k-NN classifier with the parametric model using metrics such as accuracy, confusion matrix, ROC curve, and execution time .

Module 4: Feature Extraction and Selection	**No. of Hours: 05+06**

Reducing Dimensionality: Pre-processing, Feature Selection: Inter/Intra-class Distance, Subset Selection. Feature Extraction: Principal Component Analysis, Linear Discriminant Analysis.

Textbook 1 – Ch 7

Laboratory Components

7. Write a program to perform the following tasks:
   a. Load a dataset (e.g., the Iris dataset or any relevant dataset).
   b. Preprocess the data by handling missing values and encoding categorical variables as necessary.
   c. Implement a function to calculate inter-class and intra-class distances for the features:
      - Calculate the mean feature vector for each class (intra-class).
      - Calculate the overall mean feature vector (inter-class).
      - Compute the distance metrics (e.g., Euclidean distance) to measure the separability of the classes.
   d. Select the top features based on the calculated distances and create a new dataset with these selected features.
   e. Train a classifier (e.g., logistic regression or decision tree) using the original dataset and the reduced dataset and compare their performance (e.g., accuracy, F1-score).
8. Write a program to perform the following tasks:
   a. Load a dataset (e.g., the MNIST dataset or any relevant dataset).
   b. Preprocess the data by normalizing the features and encoding categorical variables if needed.
   c. Implement Principal Component Analysis (PCA) to reduce the dimensionality of the dataset:
      - Compute the covariance matrix, eigenvalues, and eigenvectors.
      - Project the original data onto the selected principal components.
   d. Implement Linear Discriminant Analysis (LDA) for dimensionality reduction:
      - Calculate the within-class and between-class scatter matrices.
      - Compute the eigenvectors and project the data onto the new LDA space.
      - Visualize the results using scatter plots for both PCA and LDA, showing how the data is distributed in the reduced dimensions.

Module 5: Estimating and Comparing Classifiers	**No. of Hours: 05 +04**

Bias and Variance, Cross-Validation and Resampling Methods: The Holdout Method, k-Fold Cross-Validation, Bootstrap, Measuring Classifier Performance, Comparing Classifiers: ROC Curves. Combining Classifiers.

Textbook 1- Ch 9

Laboratory Components

9. Write a program to perform the following tasks:
   a. Load a dataset (e.g., the Iris dataset or the Breast Cancer dataset or any relevant data).
   b. Pre-process the data by handling missing values and normalizing features as needed.
      - Implement k-fold cross-validation
      - Split the dataset into k folds.

   For each fold, train a classifier (e.g., Decision Tree, SVM, or Logistic Regression) on the training

data and evaluate it on the validation data.

d. Record the performance metrics (accuracy, precision, recall, F1-score) for each fold.
   - Calculate the mean and standard deviation of the performance metrics across all folds to assess bias and variance.

e. Visualize the performance metrics using box plots to illustrate the variance in classifier performance.

10. Write a program to perform the following tasks:
    a. Load a binary classification dataset (e.g., the Pima Indians Diabetes dataset or any relevant dataset).
    b. Pre-process the data as necessary, including normalization and handling missing values.
    c. Train at least three different classifiers (e.g., Random Forest, Logistic Regression, and k-NN) on the dataset.
    d. For each classifier, compute the ROC curve and the Area Under the Curve (AUC):
       - Use the predicted probabilities for the positive class to compute the ROC curve.
       - Plot the ROC curve for each classifier on the same graph.
    e. Implement a simple ensemble method (e.g., Voting Classifier or Stacking) to combine the predictions of the classifiers and compute its ROC curve and AUC.
       Compare the AUC values of individual classifiers and the ensemble method. training data and evaluate it on the validation data.
    f. Record the performance metrics (accuracy, precision, recall, F1-score) for each fold.
       - Calculate the mean and standard deviation of the performance metrics across all folds to assess bias and variance.
    g. Visualize the performance metrics using box plots to illustrate the variance in classifier performance.

11. Write a program to perform the following tasks:
    a. Load a binary classification dataset (e.g., the Pima Indians Diabetes dataset or any relevant dataset).
    b. Pre-process the data as necessary, including normalization and handling missing values.
    c. Train at least three different classifiers (e.g., Random Forest, Logistic Regression, and k-NN) on the dataset.
    d. For each classifier, compute the ROC curve and the Area Under the Curve (AUC):
       - Use the predicted probabilities for the positive class to compute the ROC curve.
       - Plot the ROC curve for each classifier on the same graph.
    e. Implement a simple ensemble method (e.g., Voting Classifier or Stacking) to combine the predictions of the classifiers and compute its ROC curve and AUC.
    c. Compare the AUC values of individual classifiers and the ensemble method.

Course Outcomes:

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

CO1: Outline the fundamental concepts in pattern recognition.
CO2: Apply statistical pattern recognition techniques to solve problems.
CO3: Utilize parametric and nonparametric learning in pattern recognition.
CO4: Apply various feature extraction and selection techniques in pattern recognition.
CO5: Make use of different classifiers and compare their performance.

Textbooks:

1. Geoff Dougherty, "Pattern Recognition and Classification, An Introduction", Springer, 2012.

Reference Books:

1. Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification", 2nd Edition, John Wiley, 2006.
2. S. Theodoridis and K. Koutroumbas, "Pattern Recognition", 4th Edition, Academic Press, 2009.
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Web Links:

1. Introduction to Pattern Recognition Part I: https://www.youtube.com/watch?v=ZGUlaomeJ-k
2. Introduction to Pattern Recognition Part II: https://www.youtube.com/watch?v=RFvHa_AK5gg
3. Introduction to Pattern recognition using Machine Learning: https://www.mygreatlearning.com/blog/pattern-recognition-machine-learning/

MINI PROJECT			
Semester	**II**	CIE Marks	**50**
Course Code	**25MCSE516**	SEE Marks	**50**
Teaching Hours/Week(L:T:P)	**0:0:4**	Exam Hours	**2.5**
Total Hours	**26**	Credits	**02**

Course Learning Objectives:

This course is designed to

1. Recall the concepts learnt in Database Management Systems Course and using those concepts in application development.
2. Use the required tools and techniques to develop the mini project.
3. Examine the requirements of a mini project and transform them to a software module(s).
4. Impart skills to convert the software module to a pucca application.
5. Formulate the test cases for the project work.

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 for an ATM
3. Load Shedding in mobile systems
4. Document Security System
5. Project Planning and Management
6. Library Information System
7. College Enrolment System
8. Resilient online coverage for surveillance applications
9. Employee information and Payroll System
10. Any other application or a system

Guidelines:

- Students will carry out the mini project using DBMS and programming languages or a machine learning project or any other project with what they have learnt in the last semester or currently learning this semester.
- A team of two students should develop the mini project. However, during the examination, each student must demonstrate the project individually.
- A brief mini-project report (20-25 pages) should be submitted, the report should include Introduction, Requirements of the Mini project, Design, Implementation and Testing.
- A synopsis needs to be submitted by the team as per the specified format. The format will be shared by the department. Students should perform a prior literature search before submitting their synopsis for the mini project.
- Final evaluation will be done through project demonstration and presentation.
- The marks of the mini project would be given on the basis of performance in CIE and SEE.

Rubrics for Mini Project Evaluation CIE & SEE:

Review #	Agenda	Assessment	Review Assessment Weightage	Overall Weightage
Review 1 (R1)	Mid-Term Project Evaluation	Rubrics 1	25	50 (Sum of R1 & R2)
Review 2 (R2)	Final Internal Project Evaluation	Rubrics 2	25	
Final SEE Project Viva-Voce	End-Semester Project Evaluation		50	50
Total				100

Rubrics 1 (25 marks)

Criteria	Description	Marks (50)
Problem Definition & Relevance	Clarity of problem statement and justification	10
Background Study / Literature Review	Coverage of related work, understanding of existing methods, gaps identified	15
Objectives & Planning	Clear objectives, scope defined, feasibility plan and study and timelines	15
Preparation of Report - 1	Initial preparation of report with basic things like problem definition, related work and existing methods with references	10
	Total Marks	50

Rubrics 2 (25 marks)

Criteria	Description	Marks (50)
Technical Implementation / Methodology	Appropriateness of algorithms, coding standards, use of tools/technologies	10
Results / Intermediate Outputs	Progress achieved, partial results, testing efforts	15
Presentation & Team Contribution	Clarity in presentation/demo, Q & A, Contribution of team members	15
Preparation of Report - 2	Structure (Abstract, Introduction, Methodology, Results, Conclusion, References), formatting, flow	10
	Total Marks	50

Course Outcomes:

CO1: Apply computing concepts on the selected problem domain
CO2: Identify the requirements/objectives of the problem in hand
CO3: Design and investigate the problems to arrive at conclusions
CO4: Implement using different software tools and technologies
CO5: Technically communicate effectively as an individual/team in projects

WEB LINKS:

- https://www.youtube.com/watch?v=ejhtUjuJ_is
- https://www.youtube.com/watch?v=GdHQt8oFMsA
- https://www.youtube.com/watch?v=7v2OnUti2eM
- https://www.youtube.com/watch?v=e8g9eNnFpHQ
- https://www.youtube.com/watch?v=9rTJa4l8YQ0

III SEM

CLOUD COMPUTING			
Semester	**III**	CIE Marks	**50**
Course Code	**25MCPC601**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:0**	Exam Hours	**03**
Total Hours	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Familiarize the foundational concepts of cloud computing, service and deployment models and the essential characteristics of a cloud infrastructure.
2. Impart knowledge on core cloud services, infrastructure and virtualization.
3. Disseminate the concepts of containerization, microservices for cloud application deployment and DevOps practices.
4. Provide knowledge on cloud security challenges, Identity and Access Management, data protection and compliance strategies to secure cloud environments.
5. Familiarize advanced cloud topics including edge and fog computing, cost metrics and pricing models.

Module 1: Introduction to Cloud Computing	**No. of Hours: 10**

Overview of Cloud Computing: Definition, History, Cloud characteristics: On-Demand Usage, Ubiquitous access, Multitenancy, Elasticity, Measured Usage, Resiliency, **Cloud Delivery Models:** Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Examples and use cases of each model, Comparing Cloud Delivery Models, Combining Cloud Delivery Models, **Cloud Deployment Models:** Public, Private, Multi Clouds, Hybrid Clouds.

Textbook 1- Ch 3,4

Module 2: Cloud Infrastructure and Services	**No. of Hours: 8**

Virtualization Technology: Operating System-Based Virtualization, Hardware-Based Virtualization, Containers and Application-Based Virtualization, **Cloud Infrastructure Mechanisms:** Logical Network Perimeter, Virtual Server, Hypervisor, Cloud Storage Device, Cloud Usage Monitor, Resource Replication.

Textbook 1- Ch 5, 8
Textbook 2 - Ch 9

Module 3: Cloud Containers and its Deployment	**No. of Hours: 8**

Containerization and Microservices: Introduction to Docker and Kubernetes, Benefits and uses of containers in cloud computing. **Application Deployment on the Cloud:** CI/CD pipelines, and DevOps in cloud environments, Deploying Infrastructure as Code with CI/CD Pipelines.

Textbook 4 - Ch 1, 8, 9, 10

Module 4: Cloud Security and Compliance	**No. of Hours: 8**

Data security and storage: Aspects of data security, Data Security Mitigation, Provider data and its security, **Audit and compliance:** Internal Policy Compliance, Governance, risk and compliance.

Textbook 3 - Ch 4, 5, 8

Module 5: Emerging Trends in cloud computing	**No. of Hours: 8**

Edge and Fog Computing: Edge computing architecture, Fog computing architecture. **Cost Metrics and Pricing Models:** Business Cost Metrics, Cloud Usage Cost Metrics, Cost Management Considerations.

Textbook 1 - Ch 15, 17

Course Outcomes:

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

CO1: Apply the concepts of cloud computing by differentiating cloud characteristics, delivery models, and deployment models through real-world use cases.

CO2: Apply the concepts of virtualization techniques, cloud infrastructure components and core services to recommend solutions from major cloud providers for business needs.

CO3: Utilize Docker and Kubernetes to build, deliver, and scale containerized applications and use them in CI/CD pipelines to automate its deployment.

CO4: Make use of data security measures and compliance protocols to protect provider data, control access, and uphold governance in cloud environments.

CO5: Utilize edge and fog computing concepts for IoT use cases and assess business and cloud usage cost metrics along with cost management considerations.

Textbooks:

1. Thomas Erl and Eric Barceló Monroy, "*Cloud Computing: Concepts, Technology, Security & Architecture",* 2nd Edition, Pearson Education, 2023
2. Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi*, "Mastering Cloud Computing Foundations and Applications Programming"*, 1st Edition, Morgan Kaufmann Publishers, 2013
3. Tim Mather, Subra Kumaraswamy, and Shahed Latif, "*Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance"*, 1st Edition, O'Reilly Media, 2009
4. Mikael Krief*, "Learning DevOps",* 2nd Edition, Packt Publishing, 2022

Reference Books:

1. Michael J. Kavis, "Architecting the Cloud: Design Decisions for Cloud Computing Service Models", 1st Edition, Wiley, 2014.
2. Nigel Poulton, "*Docker Deep Dive: Zero to Docker in a Single Book"*,1st Edition, Packt Publishing, 2023.
3. Kelsey Hightower, Joe Beda, Brendan Burns, "*Kubernetes Up & Running: Dive into The Future of Infrastructure"*, 3rd Edition, Shroff publisher, 2022.

Web Links:

1. Cloud Computing Basics: https://www.youtube.com/watch?v=64-1ymY2xaw
2. Cloud Computing Full Course: https://www.youtube.com/watch?v=2LaAJq1Lb1Q
3. Cloud computing tutorial for beginners: https://www.pragimtech.com/blog/cloud/cloud-tutorial-for-beginners/
4. Introduction to Cloud Computing: https://www.coursera.org/learn/introduction-to-cloud
5. Docker Containers and Kubernetes Fundamentals: https://www.youtube.com/watch?v=kTp5xUtcalw
6. Cloud computing: https://archive.nptel.ac.in/courses/106/105/106105167/

ADVANCED JAVA			
Semester	**III**	CIE Marks	**50**
Course Code	**25MCPC602**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hours	**03**
Total Hours	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Impart fundamental knowledge and skills to build user interfaces (UIs) for Java applications.
2. Familiarize the process of retrieving, manipulating, and persisting data in a database.
3. Provide skills to utilize collections and generics effectively in Java programs.
4. Familiarize the concepts of scalable and maintainable enterprise applications using Spring Boot Framework.
5. Instill Hibernate concepts to map Java objects to relational database tables and perform CRUD operations.

Module 1: Advanced Java and Swing	**No. of Hours: 08+04**

Introduction to Advanced Java, Overview of AWT components, Event handling, Swing Overview, Basic Swing Components, Containers and Event Handling, Exploring Swing.

Textbook 1: Ch 25, 26, 32, 33

Laboratory Components

1. Build a simple GUI application using Jbutton, JtextField, and Jlabel. Set up event listeners to handle button clicks and user inputs.
2. Create an application using multiple layouts (GridLayout, BorderLayout). Implement forms and menus.
3. Build an application using Jtable, Jtree, and Dialogs. Implement data binding between GUI and backend logic.

Module 2: Database Connectivity using JDBC	**No. of Hours:08+06**

Concepts of JDBC, JDBC Process, Database Connection, Statement Objects, ResultSet, Transaction Processing, Metadata.

Textbook 2: Ch 6

Laboratory Components

4. Set up JDBC to connect to a MySQL/PostgreSQL database. Perform basic CRUD operations using CallableStatement and PreparedStatement.
5. Implement a transaction with commit and rollback in a banking or inventory application.
6. Retrieve database metadata, table structure, and execute stored procedures.

Module 3: Collections Framework and Generics	**No. of Hours: 08+04**

Introduction to Collections: Overview, The Collection Interfaces, The Collection Classes, Accessing a Collection via an Iterator, Storing User-Defined Classes in Collections, Comparators.

Generics in Java: Generics Example, Implementing Generic classes and methods, Using Wildcard Arguments, Generic Interfaces, Type Inference with Generics

Textbook 1: Ch 14, 20

Laboratory Components

7. Implement programs that demonstrate the usage of List, Set, and Map with different data types.
8. Develop a simple Generic class and perform operations using Collections with Generics.

Write a program that uses Comparable and Comparator for custom sorting, using a PriorityQueue.

Module 4: Spring Boot for Enterprise Java Applications	**No. of Hours: 08+04**

Introduction to Spring Boot, Spring Boot REST API Development, Spring Boot Data Access with JPA, Application Development with Spring MVC, Reactive Programming.

Textbook 3: Ch 1, 3, 6, 7, 8

Laboratory Components

10. Create a basic Spring Boot project and develop REST APIs.
11. Add exception handling and logging to the Spring Boot application for better error management and traceability.

Module 5: Hibernate ORM for Object-Relational Mapping	**No. of Hours: 08+06**

Introduction to Hibernate, Hibernate configuration, Mapping collections and entity associations, Advanced entity association mappings, Creating and executing queries, Hibernate Query Language (HQL).

Text book 4: Ch 2 ,7, 8, 14, 15

Laboratory Components

12. Set up Hibernate with MySQL/PostgreSQL and perform basic CRUD operations.

Course Outcomes:

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

CO1: Design and implement user-friendly graphical user interfaces using Java extensions.

CO2: Apply database operations such as data retrieval, manipulation, and storage by integrating Java with databases.

CO3: Make use of Java Collections to store, sort, and search data also apply Generics to ensure type safety.

CO4: Design simple and scalable enterprise-level applications by utilizing the Spring Boot framework.

CO5: Apply Hibernate ORM to map Java objects to relational database tables and perform CRUD operations effectively.

Textbooks:

1. Herbert Schildt and Dr. Danny Coward, "*Java: The Complete Reference*", 13th Edition, McGraw-Hill, 2024
2. Jim Keogh, "*J2EE: The Complete Reference*", McGraw-Hill, 2017
3. Mark Heckler, "*Spring Boot: Up and Running – Building Cloud Native Java and*

Kotlin Applications", O'Reilly Media, 2021
4. Christian Bauer, Gavin King and Gary Gregory, *"Java Persistence with Hibernate"*, 2nd Edition, Manning Publications, 2015

Reference Books:

1. Cay S. Horstmann, "*Core Java Volume II – Advanced Features"*, 12th Edition, Prentice Hall, 2018.
2. D.T. Editorial Services, "*Java 8 Programming Black Book"*, Dreamtech Press, 2005.
3. Catalin Tudose, "*Java Persistence With Spring Data and Hibernate"*, Manning Publications, 2023.
4. Uttam K Roy, "*Advanced Java Programming"*, Oxford University Press, 2015.

Web Links:

1. Java Database Connectivity JDBC: https://www.youtube.com/watch?v=h5sbJxeHqQY
2. Advanced Java Tutorial: https://www.edureka.co/blog/advanced-java-tutorial
3. Advanced Java Programming: https://www.youtube.com/watch?v=I_qP7H3STMg
4. Spring Boot Tutorial: https://www.youtube.com/watch?v=9SGDpanrc8U
5. Hibernate Tutorial For Beginners: https://www.youtube.com/watch?v=0KCKBv6rbkc

ETHICAL HACKING & SECURE CODING			
Semester	**III**	CIE Marks	**50**
Course Code	**25MCPE621**	SEE Marks	**50**
Teaching Hrs/Week (L: T: P)	**3:0:2**	Exam Hrs	**03**
Total Hrs	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Familiarize with the fundamental security challenges in software and to manage them throughout the development process.
2. Instill the basics of penetration tools and methodologies used in ethical hacking.
3. Impart knowledge in analyzing the vulnerabilities and attacks on a system.
4. Provide knowledge on exploitation and attacks.
5. Familiarize with the process of phishing attacks.
6. Impart knowledge on web application vulnerabilities, focusing on SQL injection.

Module 1: Security as a Software Issue, Secure Coding, Testing, Security and Software Complexity — **No. of Hrs: 10**

Introduction, Software Assurance and Software Security, Threats to Software Security, Sources of Software Insecurity, Benefits of detecting software security, What makes software secure: Defining properties of secure software, Influencing the security properties of software.
Code Analysis, Coding Practices, Software Security Testing, Security testing considerations throughout the SDLC, Security failures, Functional and Attacker's perspectives for security analysis with examples.
Textbook 3 – Ch 1, 2, 5, 6

Module 2: Introduction to Penetration Testing — **No. of Hrs: 08+06**

Stages of Penetration Test: Pre-engagement, Information gathering, Threat modeling, Vulnerability analysis, Exploitation, Post exploitation, Reporting, Kali Linux: User privileges, Data manipulation, Managing networking.
Textbook 1: Ch 0, 2, 4

Laboratory Components:

1. Perform network scanning to identify vulnerabilities and services on a target system.
2. Analyze port states and identify security implications.

Module 3: Information Gathering and Vulnerabilities — **No. of Hrs: 08+06**

Netcraft, Whois LookUps, DNS Reconnaissance, Searching for Email Addresses, Maltego. Port Scanning: Manual port scanning, Port scanning with Nmap, Finding vulnerabilities: Nessus policies, Exporting Nessus results, Researching Vulnerabilities, The Nmap Scripting Engines, Metasploit framework: Finding Metasploit modules, Setting module options, Payloads, Types of Shells, Setting a payload manually, Metasploit Scanner Modules.
Textbook 1 : Ch 5, 6

Laboratory Components:

3. Perform reconnaissance on a network, gathering publicly available information and scanning for vulnerabilities.
4. Implement advanced network scanning techniques to identify operating systems, services,

and open ports.

Module 4: Exploitation and Attacks	**No. of Hrs: 08+06**

Metasploit payloads, Exploiting WebDAV Default Credentials: Running a script on the target web server, Password attacks, Client-side exploitation: HTTP and HTTPS Payloads, Client-Side attacks, Wireless attacks: Viewing available wireless interfaces, Capturing packets, Wired equivalent privacy, WiFi Protected access, WPA2: The Enterprise Connection Process, The personal connection process, WiFi-protected setup.

Textbook 1: Ch 8, 9, 10, 15

Laboratory Components:

5. Simulate wireless attacks to test WPA/WPA2 network vulnerabilities.
6. Analyze and execute a DoS attack on a vulnerable system (in a controlled environment).

Module 5: Social Engineering and Web Application Testing	**No. of Hrs: 06+06**

The Social Engineering toolkit, Spear Phishing attacks: Choosing a payload, Single or mass mail, Creating a template, Setting up target and listener, Web attacks, Mass email attacks, Multipronged attacks, SQL Injection, Xpath Injection, Local File Inclusion, Remote File Inclusion, Command Execution, Cross-Site Scripting, Cross-Site Request Forgery, Web Application Scanning with w3af.

Textbook 1: Ch 11, T2: Ch 7

Textbook 1: Ch 14

Laboratory Components:

7. Simulate a spear phishing attack to assess the security awareness of a target organization.
8. Simulate an Email Spoofing Attack to Evaluate Email Security Measures.

Course Outcomes:

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

CO1: Use secure coding practices to build reliable and secure software applications.

CO2: Apply penetration testing techniques to identify vulnerabilities and weaknesses in systems.

CO3: Implement basic scripting to connect to a port and scan the network and host.

CO4: Perform exploitation of vulnerabilities in web, client-side, and wireless systems using penetration testing techniques.

CO5: Identify phishing attack processes and evaluate web application security by performing tests to detect vulnerabilities, with a focus on SQL injection.

Textbooks:

1. Georgia Weidman, "*Penetration Testing: A Hands-On Introduction to Hacking"*, No Starch Press, 2014
2. Daniel G. Graham, "*Ethical Hacking: A Hands-On Introduction to Breaking In",* No Starch Press, 2021
3. Julia H.Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, *Software Security Engineering : A Guide for Project Managers*, Addison-Wesley, 2008.

Reference Books:

1. Dafydd Stuttard, Marcus Pinto, "*The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws"*, Wiley Publishing, 2nd Edition, 2011
2. Michael T. Simpson, Kent Backman, and James E. Corley, *"Hands-On Ethical Hacking and Network Defense"*, 2nd Edition, Cengage Learning, 2011

Web Links:

1. Ethical Hacking (Udemy): https://www.udemy.com/course/learn-ethical-hacking-and-penetration-testing-online
2. Ethical Hacking Essentials: https://www.coursera.org/learn/ethical-hacking-essentials-ehe
3. Ethical Hacking(edx): https://www.edx.org/learn/ethical-hacking
4. Secure Software Design: https://www.udemy.com/course/secure-software-design-secure-software-series-course-3-of-8/?cou ponCode=LETSLEARNNOWPP
5. Secure System Analysis and Design: https://www.futurelearn.com/courses/system-analysis-and-design-sc

BLOCKCHAIN TECHNOLOGY AND CYBER INTELLIGENCE			
Semester	**III**	CIE Marks	**50**
Course Code	**25MCPE622**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hours	**03**
Total Hours	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Impart fundamental knowledge of Blockchain Technologies and its architecture.
2. Familiarize the concepts of decentralization in blockchain using consensus algorithms.
3. Instill the knowledge on development and deployment of smart contracts on the Ethereum blockchain.
4. Analyze cyber threats using intelligence techniques.
5. Integrate blockchain with cyber intelligence for secure systems.

Module 1: Fundamentals of Blockchain	**No. of Hours: 06+06**

The growth of blockchain technology, Distributed systems, History of blockchain and Bitcoin, Blockchain Architecture, Generic elements of a blockchain, Benefits and limitations of blockchain, types of blockchain.

Textbook 1: Ch 1

Laboratory Components:

1. Design a simple blockchain using Python or any other language of your choice that simulates the creation of blocks and chain formation.
2. Implement a simple blockchain that demonstrates the core generic elements: blocks, hash, nonce, previous hash, and transactions. Use python to simulate the creation, mining, and linking of blocks.

Module 2: Decentralization in Blockchain	**No. of Hours: 06+06**

Methods of decentralization, Quantifying decentralization, Benefits of decentralization, Full-eco system decentralization, Decentralization in practice, Routes to decentralization, Cryptographic constructs used in blockchain, Consensus algorithms: CFT (Crash Fault Tolerance) algorithm, BFT (Byzantine Fault Tolerance) algorithm.

Textbook 1: Ch 2, 5

Laboratory Components:

3. Implement a small-scale decentralized application (DApp) using blockchain to demonstrate decentralization in practice.
4. Simulate the working of a consensus algorithm (CFT) using blockchain.

Module 3: Bitcoins, Transaction Processing and Smart Contracts	**No. of Hours: 08+06**

Private and public keys in Bitcoin, Typical Bitcoin addresses, Transactions, Bitcoin payments, Bitcoin in practice, Alternative Coins, Bitcoin limitations, Smart Contracts: Definition, Ricardian contracts, Smart contract templates, Deploying smart contracts, Ethereum keys and addresses, Accounts, Transaction and messages, Ethereum blockchain, Elements of Ethereum blockchain.
Textbook 1: Ch 6, 7, 8, 9

Laboratory Components

5. Simulate a Bitcoin payment by verifying a chain of transactions (inputs and outputs) and ensuring that the total inputs match the total outputs. Example:
A sends 1 BTC to B.
B uses this 1 BTC to send 0.5 BTC to C and 0.5 BTC back to B (change output).
6. Verify each transaction in the chain.
Develop and deploy a simple smart contract using Solidity on the Ethereum test network.

Module 4: Cyber Intelligence Fundamentals and Lifecycle	**No. of Hours: 08+06**

Introduction to cyber intelligence, Types: OSINT, HUMINT, SIGINT, Intelligence lifecycle (Planning → Collection → Analysis → Dissemination), Threat intelligence sources and frameworks.
Textbook 2: Ch 1,2.

Laboratory components

7. Perform OSINT-based intelligence gathering on a suspicious organization
Scenario:
A company receives phishing emails from a domain xyz-corp-support.com. You are assigned to gather intelligence about this domain.
Task:
- Perform OSINT to identify:
- Domain ownership (WHOIS)
- Hosting details (IP, location)
- Associated emails/subdomains
- Reputation (VirusTotal / AbuseIPDB)

8. Perform intelligence cycle implementation for a ransomware attack scenario
Scenario:
An organization suspects a ransomware group targeting its network.
Task:
Apply the intelligence cycle:
- Define intelligence requirements (What to find?)
- Collect data (threat reports, IOCs)
- Process (organize indicators)
- Analyze (identify attack pattern)
- Produce a short intelligence report

Module 5: Threat Analysis, Active Defense and Integration	**No. of Hours: 06+06**

Indicators of Compromise (IoCs), Cyber Kill Chain and F3EAD model, OPSEC concepts, Blockchain for cybersecurity (secure logging, integrity), Threat intelligence collaboration.
Textbook 2: Ch 3, 4, 5, 6, 7

Laboratory Components

9. Perform OPSEC analysis on a corporate data leakage scenario
Scenario:
Sensitive company data was leaked via an employee's social media activity.
Task:
Identify:
- Critical information exposed
- Threat actors involved
- Vulnerabilities exploited
- OPSEC failures
- Recommend mitigation strategies

5. Perform IOC analysis and threat correlation using threat intelligence platforms
Scenario:
You are given suspicious indicators:
- IP: 192.168.x.x (example)
- Hash value
- Domain name

Task:
Use tools (VirusTotal, ThreatFox, etc.) to:
- Check reputation
- Identify malware family
- Correlate indicators
- Suggest response actions

Course Outcomes:
At the end of the course, the student will be able to:
CO1: Summarize the importance of distributed system's principles, blockchain technologies and bitcoins emphasizing their societal and economic significance.
CO2: Apply decentralization methods in blockchain ecosystems using cryptographic constructs and consensus algorithms.
CO3: Utilize the smart contract components and apply knowledge to develop and deploy contracts on the Ethereum blockchain
CO4: Apply cyber intelligence lifecycle for threat analysis**.**
CO5: Integrate blockchain with cyber intelligence for secure systems

Textbooks:
1. Imran Bashir, "*Mastering Blockchain - Inner Workings of blockchain, from cryptography and decentralized identities, to DeFI, NFTs and Web3",* 4th Edition, Packt, 2024.
2. Wilson Bautista Jr., "*Practical Cyber Intelligence: How action-based intelligence can be an effective response to incidents"*, Packt Publishing, 2018

Reference Books:
1. Kumar Saurabh, Ashutosh Saxena, "*Blockchain Technology Concepts and*

Applications", Wiley India Pvt. Ltd, 1st Edition, 2023.

2. Aaron Roberts, "*Cyber Threat Intelligence: The No-Nonsense Guide for CISOs and Security Managers"*, Apress, 2021

Web Links:

1. De-centralization in blockchain: https://www.angelone.in/knowledgcenter/cryptocurrency
2. Introduction to Ethereum : https://www.investopedia.com/terms/e/ethereum.asp
3. Smart Contracts: https://docs.soliditylang.org/en/latest/introduction-to-smart-contracts/
4. Threat Intelligence, Why It Matters: https://www.paloaltonetworks.com/cyberpedia/what-is-cyberthreat-intelligence-cti
5. Cyber Threat Intelligence: https://arcx.io/courses/cyber-threat-intelligence-101

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING			
Semester	III	CIE Marks	50
Course Code	25MCPE623	SEE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	Exam Hours	03
Total Hours	64 (40 Theory + 24 Lab)	Credits	04

Course Learning Objectives:
This course is designed for students to

1. Provide strong foundations in neural networks and deep learning architectures.
2. Familiarize students with optimization, regularization and representation learning techniques.
3. Demonstrate the application of CNNs, RNNs and Transformers for real-world problems.
4. Impart knowledge on Natural Language Processing fundamentals including text preprocessing, language modelling and sequence learning.
5. Enable students to design intelligent systems for vision, text analytics and language generation tasks.

Module 1 : Foundations of Deep Learning	No. of Hours: 08 + 06

Introduction to Artificial Neural Networks, Biological Neuron, Perceptron, Multilayer Feed Forward Networks, Activation Functions, Loss Functions, Gradient Descent, Backpropagation, Hyperparameters: Learning Rate, Momentum, Batch Size, Epochs, Weight Initialization, Overfitting and Regularization (Dropout, L1, L2).
Textbook 1 – Ch 6, Ch 7, Ch 8
Textbook 3 – Ch 1, Ch 2

Laboratory component

1. Build a feedforward neural network using NumPy / TensorFlow for classification problem.
2. Compare different activation functions on model accuracy.

Module 2: Deep Neural Networks	No. of Hours: 08 + 06

Deep Feedforward Networks, Optimization Methods (SGD, Adam, RMSProp), Batch Normalization, Representation Learning, Transfer Learning, Autoencoders, Denoising Autoencoders.
Convolutional Neural Networks: Convolution Layer, Pooling Layer, Padding, Stride, Fully Connected Layers, Popular CNN Architectures.
Textbook 1 – Ch 8, Ch 9, Ch 14, Ch 15
Textbook 3 – Ch 3, Ch 4

Laboratory component

3. Implement deep neural network using regularization techniques.
4. Build an autoencoder for dimensionality reduction.

Module 3: Sequential Deep Learning Models	No. of Hours: 08 + 06

Sequential Data Modelling, Recurrent Neural Networks, Vanishing Gradient Problem, LSTM, GRU, Bidirectional RNN, Sequence-to-Sequence Models, Attention Mechanism, Introduction to Transformers, Applications in speech, time-series and text.
Textbook 1 – Ch 10
Textbook 2 – Ch 8, Ch 9, Ch 10
Textbook 3 – Ch 5

Laboratory component

5. Build LSTM model for sentiment or time-series forecasting.
6. Develop simple attention-based sequence model.

Module 4 : Fundamentals of Natural Language Processing	No. of Hours: 08 + 06

Introduction to NLP, Challenges in NLP, Text Preprocessing, Tokenization, Stop-word Removal, Stemming,

Lemmatization, Part-of-Speech Tagging, Named Entity Recognition, Syntax and Parsing, N-gram Language Models, Word Embeddings: Word2Vec.
Textbook 2 – Ch 1, Ch 2, Ch 3, Ch 5, Ch 6, Ch 7

Laboratory component
7. Perform preprocessing on text corpus.
8. Implement POS tagging and Named Entity Recognition.

Train Word2Vec embeddings on sample corpus.

Module 5 : Deep Learning for NLP Applications	**No. of Hours: 08 + 06**

Neural Language Models, RNN/LSTM for Text Classification, Transformer Architecture, BERT and Pretrained Language Models, Machine Translation, Text Summarization, Question Answering, Chatbots, Sentiment Analysis, Information Retrieval using embeddings.
Textbook 2 – Ch 10, Ch 11, Ch 12, Ch 13
Textbook 3 – Ch 6, Ch 7

Laboratory component
9. Build sentiment analysis model using LSTM / BERT.
10. Implement text summarization or chatbot system.

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

CO1: Outline the mathematical foundations and architectures of deep learning models.
CO2: Apply CNNs, Autoencoders and transfer learning for vision applications.
CO3: Utilize RNN, LSTM, Attention and Transformer models for sequential data.
CO4: Apply NLP techniques for text preprocessing, parsing and language representation.
CO5: Design intelligent NLP systems using deep learning for classification, translation and summarization tasks.

TEXT BOOKS
1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
2. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition.
3. Josh Patterson and Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly, 2017.

REFERENCE BOOKS
1. Deng & Yu, Deep Learning: Methods and Applications, 1st edition, Now Publishers, 2013
2. DouweOsinga, CookBook Deep Learning 1sted, O'Reilly, 2017
3. Tanveer Siddiqui, U.S. Tiwary, **Natural Language Processing and Information Retrieval,** Oxford University Press.

WEB LINKS:
1. Foundation of Deep Learning: https://hadrienj.github.io/posts/Deep-Learning-Book-Series-2.1-Scalars-Vectors-Matrices-and-Tensors/
2. Deep Neural Networks: https://cedar.buffalo.edu/~srihari/CSE676/
3. Autoencoders: https://en.wikipedia.org/wiki/Autoencoder
4. NLP techniques for text preprocessing: https://www.nltk.org/
5. Deep Learning for NLP**:** https://spacy.io/

COMPUTER VISION FOR IMAGE AND VIDEO ANALYTICS			
Semester	III	CIE Marks	50
Course Code	25MCPE624	SEE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	Exam Hours	03
Total Hours	64 (40 Theory + 24 Lab)	Credits	04

Course Learning Objectives:
This course is designed for students to

6. Impart knowledge on fundamentals of computer vision and digital image formation.
7. Familiarize students with image preprocessing, enhancement and feature extraction techniques.
8. Provide knowledge on object detection, recognition and scene understanding methods.
9. Demonstrate image analytics techniques using deep learning models.
10. Apply video analytics methods for surveillance, tracking and activity recognition in real-time scenarios..

Module 1 : Foundations of Computer Vision and Image Formation	**No. of Hours:08 + 06**

Introduction to Computer Vision, Applications of Computer Vision, Human Vision vs Machine Vision, Image Representation, Digitization, Sampling and Quantization, Color Models, Digital Camera Pipeline, Image Formation Geometry, 2D and 3D Transformations, Perspective Projection, Lens Distortion, Image Compression Basics.
Textbook 1: Ch 1, Ch 2
Textbook 2: Ch1, Ch 2

Laboratory component
11. Perform scaling, rotation, translation and perspective correction.
12. Implement histogram analysis and image compression comparison.

Module 2: Image Processing and Feature Extraction	No. of Hours: 08 + 06

Point Operations, Contrast Stretching, Histogram Equalization, Spatial Filtering, Smoothing, Sharpening, Edge Detection (Sobel, Prewitt, Canny), Frequency Domain Processing using Fourier Transform, Image Restoration, Feature Detection and Description (Harris, SIFT, ORB), Feature Matching.
Textbook 1: Ch 3, Ch4
Textbook 2: Ch3, Ch4, Ch 5

Laboratory component
13. Implement edge detection using Canny and Sobel operators.
14. Detect and match features between two images.

Module 3: Image Segmentation and Object Recognition	No. of Hours: 08 + 06

Image Segmentation: Thresholding, Region Growing, Watershed, Clustering based segmentation. Contours and Shape Analysis. Object Detection Concepts, Bounding Boxes, IoU, Non-Maximum Suppression. Deep Learning based Detection: R-CNN, Fast R-CNN, Faster R-CNN, YOLO. Face Detection and Face Recognition, Scene Understanding.
Textbook 1: Ch 5, Ch 14
Textbook 3: Ch 4, Ch 5

Laboratory component
15. Detect objects using pretrained YOLO model.
16. Build a simple face recognition system.

Module 4 : Image Analytics	No. of Hours: 08 + 06

Introduction to CNNs, Convolution, Pooling, Transfer Learning, Image Classification, Image Retrieval, Image Captioning Basics, Medical Image Analytics, Defect Detection, Gesture Recognition, Facial

Expression Analysis, Model Evaluation Metrics.
Textbook 3: Ch 2, Ch3, Ch 6
Textbook 1: Ch 14

Laboratory component
17. Apply transfer learning using ResNet / MobileNet.
18. Implement image similarity retrieval system.

Module 5 : Video Analytics and Intelligent Surveillance	No. of Hours: 08 + 06

Video Fundamentals, Frames and Temporal Information, Motion Detection, Background Subtraction, Optical Flow, Object Tracking, Multi-object Tracking, Action Recognition, Event Detection, Video Summarization, Deep Learning Architectures for Video Analytics (3D CNN, CNN-LSTM), Smart Surveillance Applications.
Textbook 1: Ch 8, Ch 9
Textbook 3: Ch 7

Laboratory component
19. Detect moving objects using background subtraction.
20. Implement object tracking in video.

Course Outcomes:
At the end of the course, the student will be able to:
CO1: Outline image formation, representation and geometric transformations in computer vision.
CO2: Apply image enhancement, filtering and feature extraction techniques.
CO3: Utilize segmentation and object recognition methods for practical applications.
CO4: Design deep learning models for image analytics tasks.
CO5: Develop intelligent video analytics systems for tracking, surveillance and activity recognition..

TEXT BOOKS
4. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2020.
5. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis and Machine Vision, Cengage Learning.
6. Vaibhav Verdhan, Computer Vision Using Deep Learning, Apress, 2021.

REFERENCE BOOKS
4. D. Forsyth and J. Ponce, Computer Vision: A Modern Approach, Pearson.
5. E. R. Davies, Computer and Machine Vision, Academic Press.
6. Caifeng Shan et al., Video Analytics for Business Intelligence, Springer.
7. Rafael Gonzalez and Richard Woods, Digital Image Processing, Pearson.

WEB LINKS:
6. Introduction to Computer Vision: https://www.geeksforgeeks.org/computer-vision
7. Practical applications: https://www.kaggle.com/learn/computer-vision
8. Deep learning models: https://www.analyticsvidhya.com/blog/2023/02/deep-residual-learning-for-image-recognition-resnet-explained/

MAJOR PROJECT			
Semester	**III**	CIE Marks	**100**
Course Code	**25MCSE603**	SEE Marks	**-**
Teaching Hours/Week (L:T:P)	**-**	Exam Hours	**-**
Total Hours	**-**	Credits	**08**

Course Learning Objectives:

This course is designed to

1. Learn how to gather pertinent information from different sources and learn to perform a good literature survey.
2. Empower a student's ability to apply concepts, theories, and methodologies from their coursework to solve real-world problems or meet specific project requirements.
3. Impart latest industry requirements and get skilled in those areas to deliver solutions for problems.
4. Enable a student to improve his/her presentation skills and time management skills.
5. Gather the requirements of the project work and present his/her work in a technical seminar and documenting the project work via a research thesis.

Guidelines

- A student will be allotted a supervisor/guide in the department.
- He/She in consultation with the guide has to carry out a proper literature survey/visit industry to finalize the topic of the major project.
- Developing a Software Requirements Specification (SRS) document.
- Carry out the project work in regular consultation with the supervisor/guide.
- Attend Project reviews.

Course Outcomes:

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

CO1: Use prior art search literature and arrive at the right problem definition.
CO2: Apply disciplined software engineering practices to collect the requirements for the given problem.
CO3: Build sound technical knowledge in his/her selected project topic and domain.
CO4: Use critical thinking and problem-solving skills required for developing a software application/solution.
CO5: Choose and acquire the appropriate skills required for a typical project from time to time.

IV SEM

INTERNSHIP			
Semester	**IV**	CIE Marks	**50**
Course Code	**25MCSE611**	SEE Marks	**50**
Teaching Hours/Week (L: T:P)	**-**	Exam Hours	**03**
Total Hours	**-**	Credits	**10**

Course Learning Objectives:

1. To realize the real-time experience of the software industry.
2. Identify and understand the importance of working as a team in a company.
3. To gain professional skills and learn to work independently towards a variety of projects.
4. Learn and adhere to professional standards in the industry.

Guidelines

- Students under the guidance of internal guide/s and external guide shall take part in all the activities regularly to acquire as much knowledge as possible without causing any inconvenience at the place of internship. The duration of the internship will be for entire semester. This is a mandatory internship that should be completed by a student during the re semester.
- Once the student completes his/her internship, the student is required to present a seminar on the internship to the Internship Coordinator after consultation with the Head of the Department.
- A report of the internship has to be prepared and submitted during the review seminar.

Course Outcomes:

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

CO1: Apply the knowledge and skills learnt during their internship in their project work.
CO2: Identify the different technical skills required to a variety of project requirements from time to time.
CO3: Identify the different areas that can be explored depending on one's interest and skill levels.
CO4: Develop a better understanding on the career options available in the industry today.

MOOC			
Semester	**IV**	CIE Marks	*
Course Code	**25MCSE612**	SEE Marks	*
Teaching Hours/Week (L:T:P)	-	Exam Hours	-
Total Hours	-	Credits	**03**

Guidelines:

- Every student should mandatorily complete one NPTEL course of 8 weeks duration during their MCA program. Three credits will be given on successful completion of the NPTEL course.
- He/She can take up the course in their respective stream of interest already opted before the beginning of the second/third semester or any other computer science/computer applications/computer engineering course.
- The BoS recommended course list will be shared with all the students and the student is expected to complete a course that is available in the list. This will be announced once the NPTEL course list is out in the Swayam portal. A student should select a course only from the BoS recommended list.
- For those students who did not clear the NPTEL course during second/third semester, the department will offer a course in fourth semester in online mode for which CIE is 50 marks and SEE is 50 marks will be conducted by the department. The student will undergo the same process to study the course online as conducted by the online portal. The examination will be conducted by the department in the similar way as a professional elective course, which would be evaluated based on the CIE, ABA and SEE marks.

PUBLICATION / PATENTING			
Semester	IV	CIE Marks	100
Course Code	**25MCSE613**	SEE Marks	-
Hours/Week (L: T: P)	-	Exam Hours	-
Total Hours	-	Credit	04

Course Learning Objectives: This course is designed to

1. Enable students to systematically document literature review and research gaps.
2. Train students in research paper writing and publication process.
3. Provide knowledge of patentability search and drafting patent applications.
4. Motivate students to publish research work or file patents.
5. Inculcate ethical research practices.

Procedure for CIE:

1. The project guide from Major Project shall continue as the research guide, ensuring continuity in converting the completed project work into a research publication or patent.
2. Students shall derive research contributions from their completed project work, focusing on:
   a. Accurately representing experimental results
   b. Providing evidence for validated outcomes / effective performance
   c. Explicit mention of novelty / innovation
3. The evaluation shall be carried out based on the following three major components:
   Component 1: Literature Review & Manuscript / Patent Drafting
   a. Comprehensive literature survey and identification of research gap
   b. Structuring of research paper / patent document
   c. Technical depth, methodology, and clarity of presentation
   d. Patentability analysis (in case of patent track)

   Component 2: Submission / Filing
   a. Submission of manuscript to a reputed SCI/WoS indexed journal
   OR
   Filing of a patent application with appropriate authority
   b. Submission proof (acknowledgement / application number) is mandatory

   Component 3: Publication / Patent Outcome
   Publication of article in SCI/WoS indexed journal
   OR
   Publication of patent
4. Marks shall be awarded proportionately based on:
   a. Paper publication status (accepted / published)
   b. Patent filing / publication status
5. Students must ensure originality and adherence to ethical practices, including proper citation and avoidance of plagiarism.
6. A plagiarism report shall be submitted, with similarity generally not exceeding 10%, failing which revision and resubmission is required.
7. Progress shall be monitored through periodic reviews by the departmental committee, focusing on:
   a. Conversion of project work into quality manuscript / patent

b. Quality of manuscript / patent drafting
c. Readiness for submission / publication

8. The HOD shall constitute publication/patenting evaluation/review committee(s) & the composition shall be as follows:
    a. HOD or one of the HODs in case of an interdisciplinary project, shall be the Chairman of the committee.
    b. Coordinator shall be member - Convener
    c. Guide shall be the member
    d. One/Two senior faculty members nominated by the HOD (may be from different departments in case of an interdisciplinary project jointly nominated by the HODs)

Deliverables:

1. Literature Review Report / Patentability Report
    a. Representation of problem identification
    b. Representation of gap analysis
    c. Survey of recent literature / prior art
2. Manuscript / Patent Draft
    Structured manuscript (Abstract, Introduction, Methodology, Results, References)
    OR
    Patent Draft (Title, Abstract, Claims, Description, Drawings)
3. Submission Proof
    Journal submission acknowledgement / Patent filing receipt
4. Paper Publication / Patent Filing / Patent Publication Evidence

Evaluation for CIE:

Total of 100 CIE marks is distributed as follows:

Sl. No.	Description	Marks
1.	Literature Review, Research Paper Writing / Patentability Search, Drafting the Patent Application	50
2.	a. Submission & Acceptance of manuscript (SCI/WoS/Scopus) OR b. Filing a Patent Application	20
3.	a. Publication of Article (SCI/WoS/Scopus) OR b. Publication of Patent	30
	Total	**100**

Course Outcomes:

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

CO1: Systematically documenting literature review and gaps in a specific domain.
CO2: Prepare a structured research manuscript or patent document following standard guidelines.
CO3: Perform patentability search and analyze prior art for innovation feasibility.
CO4: Demonstrate the ability to submit research work to journals or file patent applications.
CO5: Apply ethical practices in research, including plagiarism avoidance and proper citation.