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

SCHEME & SYLLABUS
I/II/III/IV SEMESTER MCA PROGRAM

2023 Scheme

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

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.*

MITE
Invent Solutions

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	23MCPC511	Mathematical Foundation for Computer Applications	**I**
2	23MCPC512	Data Structures	**I**
3	23MCPC513	Database Management Systems	**I**
4	23MCPC514	Web Technologies	**I**
5	23MCPC515	Design and Analysis of Algorithms	**I**
6	23MCPC516	Software Engineering	**I**
7	23MCPC521	Operating System	**II**
8	23MCPC522	Object Oriented Programming with Java	**II**
9	23MCPC523	Computer Networks	**II**
10	23MCPC524	Python Programming	**II**
11	23MCPC611	Cloud Computing	**III**
12	23MCPC612	Advanced Java	**III**
13	23MCPC613	Big Data Analytics	**III**
14	23MCPC614	Object Oriented Modeling and Design	**III**
PROFESSIONAL ELECTIVE (PE)			
15	23MCPE551	Cryptography and Cyber Security	**II**
16	23MCPE552	Network Security	**II**
17	23MCPE553	Machine Learning	**II**
18	23MCPE554	Natural Language Processing	**II**
19	23MCPE651	Ethical Hacking	**III**
20	23MCPE652	Digital Forensics	**III**
21	23MCPE653	Deep Learning	**III**
22	23MCPE654	Pattern Recognition	**III**
23	23MCPE661	Secure Software Development	**IV**
24	23MCPE662	Blockchain Technology	**IV**
25	23MCPE663	Computer Vision	**IV**
26	23MCPE664	Applications of Machine Learning for Image and Video Analytics	**IV**
27	23MCPE671	Disaster Management and Business Continuity in Cybersecurity	**IV**
28	23MCPE672	Cyber Intelligence	**IV**
29	23MCPE673	Augmented Reality and Virtual Reality	**IV**

30	23MCPE674	Reinforcement Learning	**IV**
HUMANITIES (HM)			
31	23MCHM525	Research Methodology & IPR	**II**
SKILL ENHANCEMENT (SE)			
32	23MCSE526	Mini Project	**II**
33	23MCSE615	Major Project Phase - 1	**III**
34	23MCSE616	Internship	**III**
35	23MCSE621	Major Project Phase - 2	**IV**
NO CREDIT MANDATORY(NM)			
36	23MCNM517	Basics of Computer Programming	**I**
37	23MCNM622	MOOC Course	**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	23MCPC511	Mathematical Foundation for Computer Applications	PC	Mathematics	4	0	0	50	50	100	3	4
2	23MCPC512	Data Structures	PC	MCA	3	0	2	50	50	100	3	4
3	23MCPC513	Database Management Systems	PC	MCA	3	0	2	50	50	100	3	4
4	23MCPC514	Web Technologies	PC	MCA	4	0	0	50	50	100	3	4
5	23MCPC515	Design and Analysis of Algorithms	PC	MCA	4	0	0	50	50	100	3	4
6	23MCPC516	Software Engineering	PC	MCA	4	0	0	50	50	100	3	4
7	23MCNM517	Basics of Computer Programming*	*NM	MCA	2	0	2	100	-	100	3	-
											Total	24

*NM – No Credit Mandatory Course

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

II 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	23MCPC521	Operating System	PC	MCA	4	0	0	50	50	100	3	4
2	23MCPC522	Object Oriented Programming with Java	PC	MCA	3	0	2	50	50	100	3	4
3	23MCPC523	Computer Networks	PC	MCA	4	0	0	50	50	100	3	4
4	23MCPC524	Python Programming	PC	MCA	2	0	2	50	50	100	3	3
5	23MCHM525	Research Methodology & IPR	HM	MCA	4	0	0	50	50	100	3	4
6	23MCPE55X	Professional Elective - 1	PE	MCA	*-	*-	*-	50	50	100	3	4
7	23MCSE526	Mini Project	SE	MCA	0	0	4	50	50	100	3	2
											Total	25

*Professional Elective-based structure

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	23MCPE551	Cryptography and Cyber Security	PE	MCA	4	0	0	50	50	100	3	4
2	23MCPE552	Network Security	PE	MCA	4	0	0	50	50	100	3	4

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	23MCPE553	Machine Learning	PE	MCA	3	0	2	50	50	100	3	4
2	23MCPE554	Natural Language Processing	PE	MCA	3	0	2	50	50	100	3	4

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

III 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	23MCPC611	Cloud Computing	PC	MCA	4	0	0	50	50	100	3	4
2	23MCPC612	Advanced Java	PC	MCA	3	0	2	50	50	100	3	4
3	23MCPC613	Big Data Analytics	PC	MCA	3	0	2	50	50	100	3	4
4	23MCPC614	Object Oriented Modeling and Design	PC	MCA	2	0	2	50	50	100	3	3
5	23MCPE65X	Professional Elective - 2	PE	MCA	*-	*-	*-	50	50	100	3	3
6	23MCSE615	Major Project Phase - 1	SE	MCA	-	-	-	100	-	100	-	5
7	23MCSE616	Internship	SE	-	10 weeks internship completed during the intervening vacation of II & III Semester			100	-	100	3	6
											Total	29

*Professional Elective-based structure

Professional Elective – 2

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	23MCPE651	Ethical Hacking	PE	MCA	2	0	2	50	50	100	3	3
2	23MCPE652	Digital Forensics	PE	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	23MCPE653	Deep Learning	PE	MCA	2	0	2	50	50	100	3	3
2	23MCPE654	Pattern Recognition	PE	MCA	2	0	2	50	50	100	3	3

IV 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	23MCPE66X	Professional Elective - 3	PE	MCA	*-	*-	*-	50	50	100	3	3
2	23MCPE67X	Professional Elective - 4	PE	MCA	*-	*-	*-	50	50	100	3	3
3	23MCSE621	Major Project Phase - 2	SE	-	-	-	-	100	100	200	3	16
4	23MCNM622	Recommended MOOC Courses	*NM	-	-	-	-	-	-	-	-	-
											Total	22

*NM – No Credit Mandatory Course
*Professional Elective-based structure

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

Professional Elective – 3

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	23MCPE661	Secure Software Development	PE	MCA	3	0	0	50	50	100	3	3
2	23MCPE662	Blockchain Technology	PE	MCA	2	0	2	50	50	100	3	3

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	23MCPE663	Computer Vision	PE	MCA	2	0	2	50	50	100	3	3
2	23MCPE664	Applications of Machine Learning for Image and Video Analytics	PE	MCA	2	0	2	50	50	100	3	3

Professional Elective – 4

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	23MCPE671	Disaster Management and Business Continuity in Cyber Security	PE	MCA	3	0	0	50	50	100	3	3
2	23MCPE672	Cyber Intelligence	PE	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	23MCPE673	Augmented Reality and Virtual Reality	PE	MCA	2	0	2	50	50	100	3	3
2	23MCPE674	Reinforcement Learning	PE	MCA	2	0	2	50	50	100	3	3

I SEM

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

Course Learning Objectives:

This course is designed to

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

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

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

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

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

Module 2: Mathematical logic	**No. of Hrs: 12**

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

Self-study: Performing logical operations using MATLAB.

Module 3: Graph theory	**No. of Hrs: 12**

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

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

Module 4: Statistics	**No. of Hrs: 12**

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

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

Module 5: Probability	**No. of Hrs: 12**

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

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

Course Outcomes:

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

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

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

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

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

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 Scientists"*, 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, Macmillan Publishers India, 2018.
3. Oliver C. Ibe, "*Fundamentals of Applied Probability and Random Process"*, 2nd Edition, Elsevier Academic Press, 2023.

Web Links:

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

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

Course Learning Objectives:

This course is designed to

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

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

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

Laboratory Component

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

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

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

Laboratory Component

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

Module 3: Linked List	**No. of Hrs: 13**

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

Laboratory Component

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

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

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

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

Laboratory Component

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).

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

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

Laboratory Component

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

Course Outcomes:

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

CO1: Identify the fundamental concepts in data structures, their applications, dynamic memory management.

CO2: Apply stacks and queues to solve applications.

CO3: Implement the operations of linked lists and use it to execute stacks and queues.

CO4: Identify and use the appropriate sorting and searching techniques for a given scenario.

CO5: Build different types of trees and perform traversals on graphs.

Textbooks:

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

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.
3. Kenneth A Berman and Jerome L Paul, "*Algorithms"*, 2nd Edition, Cengage Learning India Pvt Ltd, 2002.

Web Links:

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

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

Course Learning Objectives:

This course is designed to

1. Provide a strong foundation in database concepts, technology, and practice.
2. 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 Hrs: 12**

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

Laboratory 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:

   b. List out the student details, and their course details. The records should be ordered in

semester-wise manner.

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

Module 2: Relational Model	**No. of Hrs: 13**

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

Laboratory 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 Hrs:13**

SQL:SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. **Advanced Queries:**More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.

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 Hrs: 13**

Normalization: Introduction to Normalization using Functional andMultivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies,Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form,Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Examples of Normal Forms.

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 Hrs: 13**

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.

Course Outcomes:

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

CO1: Describe database objects and integrity constraints on a database using RDBMS.
CO2: Implement SQL queries for database manipulation.
CO3: Use simple database systems to relate the concept of transaction, concurrency control and recovery.
CO4: Apply the concepts of normalization and design effective databases.

Text Books:

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

Reference Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "*Database System Concepts*", 6th Edition, Tata McGraw Hill Education Private Limited, 2011.

Web Links:

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

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

Course Learning Objectives:

This course is designed to

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

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

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

Module 2:JavaScript	**No. of Hrs: 10**

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

Module 3: Angular JS	**No. of Hrs: 10**

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

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

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

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

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

Course Outcomes:

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

CO1: Describe the concepts of CSS and XHTML5.

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

CO3: Develop dynamic web pages using AngularJS.

CO4: Demonstrate the connectivity between ReactJS code and a web server.

Textbooks:

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

Reference Books:

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

Web Links:

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

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

Course Learning Objectives:

This course is designed to

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

Module 1 : Introduction	**No. of Hrs: 12**

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

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

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

Module 3: Algorithms	**No. of Hrs: 10**

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

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

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

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

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

Course Outcomes:

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

CO1: Explain the methods of analyzing the algorithms and its performance.

CO2: Solve problems on Brute Force, Divide and Conquer algorithms and measure their performance.

CO3: Solve problems on different Decrease & Conquer algorithms and analyze their time and space complexities.

CO4: Apply the concepts of Backtracking, Branch & Bound techniques in solving problems and describe whether the given algorithm belongs to P,NP, NP-Hard or NP-Complete complexity classes .

Textbooks:

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

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

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

Course Learning Objectives:

This course is designed for students to

1. Apply software engineering principles and activities in creating software applications.
2. Distinguish between the functional and non-functional requirements of an application.
3. Build a Software Requirement Specification (SRS) document based on the requirements for a given application.
4. Identify the ethical and professional issues and infer as to why they are of concern to software engineers.
5. Design effective software engineering applications in real-time satisfying the quality perspective.

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

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

Module 2: Requirements	**No. of Hrs: 12**

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

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

Module 3: Design	**No. of Hrs: 10**

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

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

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

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

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

Module 5: Project Management	**No. of Hrs: 10**

Project management: Risk management, Managing people, Teamwork.

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

Course Outcomes:

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

CO1: Choose the right software process model for a given problem.

CO2: Build a Software Requirements Specification document for an application.

CO3: Make use of the principles of Architectural design, service engineering and software testing while developing software.

CO4: Develop a project plan to deliver a software application.

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

BASICS OF COMPUTER PROGRAMMING			
Semester	**I**	CIE Marks	**100**
Course Code	**23MCNM517**	SEE Marks	**-**
Teaching Hours/Week (L:T:P)	**-**	Exam Hrs	**03**
Total Hours	**35 (25 hrs Theory + 10 hrs Lab)**	Credits	**-**

Course Learning Objectives:

This course is designed to

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

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

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

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

Laboratory Component

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

Module 2: Structures	**No. of Hrs: 05**

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

Laboratory Component

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

Module 3: Pointers	**No. of Hrs: 05**

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

Laboratory Component

6. Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of N real numbers.
7. Write a program to find the reverse of a string using pointers.

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

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

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

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

Course Outcomes:

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

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

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

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

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

Textbooks:

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

Reference Books:

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

Web Links:

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

II SEM

OPERATING SYSTEM			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPC521**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**4:0:0**	Exam Hrs	**03**
Total Hrs	**52**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Teach the basic concepts and structure of operating systems.
2. Impart knowledge on different process scheduling techniques.
3. Make students familiar with the basics of process management and file systems.
4. Instill the basic knowledge of shell programming.

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

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

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

Module 2: Process Management	**No. of Hrs: 10**

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

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

Module 3: Synchronization and Deadlocks	**No. of Hrs: 12**

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

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

Module 4 : The File System	**No. of Hrs: 10**

The File System: The File, What's in a File name? The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, The Unix File System.

Basic File Attributes: ls options, File Ownership, File Permissions, chmod, Directory Permissions, Changing the File Ownership

More File Attributes: File Systems and Inodes, Hard Links, Symbolic Links, The Directory, umask, Modification and Access Times, find.

Module 5: Shell Programming	**No. of Hrs: 10**

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.

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 to perform different operations.

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

CO3: Summarize the different system calls used in process management and file management.

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

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.
4. Harvey M Deital, "*Operating Systems",* 3rd Edition, Addison Wesley, 1990.

Web Links:

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

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

Course Learning Objectives:

1. Recognize the importance of the fundamental concepts in 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. Apply exception handling techniques on typical programming problems.
5. Build applications using the concepts of multithreading.

Module 1: OOPS Concepts and Java Programming	**No. of Hrs: 08**

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

Laboratory Component

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

   1
   1 2
   1 2 3
   1 2 3 4
   1 2 3 4 5

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

Module 2: Introduction to Java	**No. of Hrs: 08**

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.

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 Hrs:08**

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

Laboratory Components

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 Hrs:08**

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

Laboratory Components

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 Hrs:08**

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

Laboratory Components

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

Course Outcomes:

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

CO1: Illustrate the basic concepts of Object Oriented Programming in Java.
CO2: Use the inheritance concepts on a variety of real-time applications.
CO3: Apply Exception handling concepts to write effective programs in Java.
CO4: Develop applications using Multithreaded Programming.

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

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

Course Learning Objectives:

This course is designed to

1. 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 Hrs: 12**

Introduction, 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.

Module 2 : Data Link Layer	**No. of Hrs: 10**

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)), The medium access control sublayer: The Channel Allocation Problem.

Module 3 : Network Layer	**No. of Hrs: 10**

Network Layer Design issues, Routing algorithms- Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing, Hierarchical routing, Broadcast Routing, Multicast Routing, Anycast Routing.

Module 4 : Transport Layer	**No. of Hrs: 10**

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

Module 5 : Application Layer	**No. of Hrs: 10**

Principles of Network Applications, Web and HTTP, Dynamic web pages and Web applications, Electronic Mail in the Internet, DNS: DNS Namespace and Hierarchy, DNS queries and responses, DNS Privacy.

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: Solve the issues related with the transport and application layers by using various networking techniques.

TEXT BOOK

1. Andrew S. Tanenbaum, David J Wetherall, "*Computer Networks"*, 6th Edition, Pearson Education, 2023.

Reference Books:

1. Behrouz A Forouzan, "*Data Communication and Networking"*, 5th Edition, Tata McGraw-Hill, 2013.
2. Alberto Leon-Garcia, Indra Widjaja, "*Communication Networks - Fundamental Concepts and Key architectures",* 2nd Edition, Tata McGraw-Hill, 2004.
3. William Stallings, "*Data and Computer Communication",* 8th Edition, Pearson Education, 2007

Web Links:

1. https://elearn.daffodilvarsity.edu.bd/course/view.php?id=5457
2. https://onlinecourses.nptel.ac.in/noc21_cs18/preview
3. https://www.geeksforgeeks.org/error-detection-in-computer-networks/
4. https://www.javatpoint.com/computer-network-tutorial

PYTHON PROGRAMMING			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPC524**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:
This course is designed to

1. Provide skills to demonstrate Python collection objects and functions.
2. Impart knowledge to implement object-oriented programming using Python.
3. Deliver working of numpy array functionalities and pandas data structures for data analysis.
4. Elaborate the steps involved in data wrangling.
5. Provide insights on data visualization using matplotlib and seaborn libraries.

Module 1: Python Basic Concepts and Programming	**No. of Hrs: 6**

Interpreter, Program Execution, Statements, Expressions, Flow Controls, Functions, Numeric Types, Sequences, Strings, Parts of Python Programming Language, Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, The type( ) Function and Is Operator, Control Flow Statements, The continue and break Statements, Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.

Laboratory Component	**No. of Hrs: 6**

1. Write a Python program to perform a linear search.
2. Write a Python program to insert an element into a sorted list.
3. Write a Python program using object-oriented programming to demonstrate encapsulation, overloading and inheritance.

Module 2: Python Collection Objects, Classes	**No. of Hrs: 6**

Strings, Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings, Lists-Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods. Sets, Tuples and Dictionaries. Files: reading and writing files. Class Definition, Constructors, Inheritance, Overloading.

Laboratory Component	**No. of Hrs: 6**

4. Implement a python program to demonstrate
    a. Importing Datasets
    b. Cleaning the Data
    c. Data frame manipulation using Pandas.
5. Implement a python program to demonstrate the following using numpy
    a. Array manipulation, Searching, Sorting and splitting.
    b. Broadcasting and Plotting numpy arrays.

6. Implement a python program to demonstrate Data visualization with various Types of Graphs using numpy.
Module 3: Introduction to Numpy and Pandas **No. of Hrs: 5**
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.
Laboratory Component **No. of Hrs: 6**
7. Write a Python program that creates a m X n integer array and print its attributes using matplotlib. 8. Write a Python program to demonstrate the generation of a. Linear regression models. b. Logistic regression models.
Module 4: Data Loading and Data Wrangling **No. of Hrs: 5**
Reading and writing data in text format, Interacting with the database, Combining and Merging data sets, Reshaping and pivoting, Data transformation, String manipulation.
Laboratory Component **No. of Hrs: 6**
9. Write a Python program to demonstrate Time series analysis with Pandas. 10. Write a Python program to demonstrate Data Visualization using Seaborn
Module 5: Visualization with Matplotlib and Seaborn **No. of Hrs: 4**
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.
Course Outcomes: At the end of the course, the student will be able to: **CO1:** Apply python programming concepts to solve the given scenarios. **CO2:** Make use of Python collection objects and functions while solving problems. **CO3:** Utilize numpy array functionalities and pandas data while creating applications. **CO4:** Apply the steps involved in data wrangling and visualize the data using matplotlib and seaborn libraries.
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 VanderPlas, "*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. https://www.w3schools.com/python/
2. https://www.geeksforgeeks.org/python-programming-language-tutorial/
3. https://docs.python.org/3/tutorial/index.html

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

Course Learning Objectives:

This course is designed to

1. Deliver an overview of the research methodology and explain the technique of defining a research problem.
2. Instil the process of carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.
3. Provide knowledge to design the research and select essential data sampling methods to solve research problems.
4. Apply the mechanisms involved in collecting data, performing analysis, interpret the outcomes and write a research paper.
5. Make students familiar with Intellectual Property Rights and its importance.

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

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

Module 2: Defining the Research Problem and Literature Survey	**No. of Hrs: 12**

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

Module 3: Research Design and Data Sampling	**No. of Hrs: 10**

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

Module 4: Data Collection, Analysis and Report writing	**No. of Hrs: 10**

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

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

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

Course Outcomes:

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

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

CO2: Choose 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 technique to solve a research problem.

CO5: Analyze 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.

Web Links:

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

Professional Electives:

CRYPTOGRAPHY AND CYBERSECURITY			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPE551**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**4:0:0**	Exam Hrs	**03**
Total Hours	**52**	Credits	**04**

Course Learning Objectives:

This course is designed for students to

1. Summarize the basic concepts of Computer security and cryptography and the different Symmetric and Asymmetric cryptographic algorithms.
2. Gain knowledge of public-key cryptography, Message Authentication Algorithms and Hash Functions.
3. Familiarize themselves with the various types of cyber-attacks and cyber-crimes and an overview on the cyber laws.
4. Discover Cyber Security challenges and implications including Cyber Crime.

Module 1 : Attacks on Computers, Computer Security and Cryptography	**No. of Hrs: 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, key range and key size, possible types of attacks.

Module 2: Cryptosystems and Algorithms	**No. of Hrs: 10**

Symmetric key Ciphers: Block Cipher principles & Algorithms (DES, AES), Block cipher modes of operation, Stream ciphers, RC4, Location and placement of encryption function, Key distribution.

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

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

Module 3: Cyber Security	**No. of Hrs: 10**

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Spectrum of attacks, Taxonomy of various attacks, IP spoofing, Methods of defense, Security Models, risk management, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy

Module 4 : Cyber Security and privacy	**No. of Hrs: 10**
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.	
Module 5 : Cybercrime and Cyber terrorism	**No. of Hrs: 12**
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.	

Course Outcomes:

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

CO1: Apply the basic cryptographic concepts to solve security issues and problems.

CO2: Use different Message Authentication Algorithms and Hash Functions in typical scenarios.

CO3: Identify the different cyber threats existing today and develop strategies to mitigate them.

CO4: Utilize the knowledge of cybercrime and cyber terrorism to analyze real-world scenarios, identifying their impacts on various societal aspects.

Textbooks:

1. Cryptography and Network Security: William Stallings, Pearson Education,411i Edition
2. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley
3. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press..

Reference Books:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, lst Edition.
2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 2"d Edition
3. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335, 2018.
4. Introduction to Cyber Security , Chwan-Hwa(john) Wu,J.David Irwin.CRC Press T&F Group

Web Links:

1. https://www.tutorialspoint.com/cryptography/index.htm
2. https://www.w3schools.com/cybersecurity/

NETWORK SECURITY			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPE552**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**4:0:0**	Exam Hrs	**03**
Total Hours	**52**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. 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 Hrs: 10**

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)

Introduction to Cryptography: Symmetric vs. Asymmetric Cryptography, Terminology (Plaintext, Ciphertext, Key), Identifying Security Threats in Real-World Scenarios.

Module 2: Cryptographic Techniques	**No. of Hrs: 10**

Symmetric Key Cryptography: Classical Encryption Techniques (Caesar Cipher, Vigenere Cipher), Modern Block Ciphers (DES, AES), Operating Modes (CBC, ECB, CTR).

Asymmetric Key Cryptography: Public Key Infrastructure (PKI), RSA Algorithm, Digital Signatures, Key Management.

Module 3: Network Security Applications	**No. of Hrs: 10**

Transport Layer Security (TLS)/Secure Sockets Layer (SSL), Secure Communication over Networks, HTTPS, Secure Shell (SSH)

Virtual Private Networks (VPNs), Tunneling Protocols (PPTP, L2TP, OpenVPN), Remote Access Security

Module 4: Network Security Threats and Mitigations	**No. of Hrs: 10**

Introduction to Network Attacks: Denial-of-Service (DoS) Attacks, Man-in-the-Middle (MitM) Attacks, Session Hijacking, Spoofing Attacks.

Intrusion Detection and Prevention Systems (IDS/IPS): Signature-based vs. Anomaly-based Detection, Firewall Technologies (Packet Filtering, Stateful Firewalls)

Module 5: Network Security Management	**No. of Hrs: 12**

Network Security Best Practices: User Authentication and Access Control, Secure Password Management, Patch Management and Vulnerability Scanning.

Network Monitoring and Analysis Tools: Wireshark, Security Information and Event Management (SIEM) Systems.

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. https://www.sans.org/white-papers/454/
2. https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:online-data- security/xcae6f4a7ff015e7d:data-encryption-techniques/a/public-key-encryption
3. https://developer.mozilla.org/en-US/docs/Web/Security/Transport_Layer_Security
4. https://securityscorecard.com/blog/securityscorecard-10-risk-factors-explained/
5. https://www.elastic.co/guide/en/siem/guide/index.html

MACHINE LEARNING			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPE553**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hrs	**03**
Total Hours	**64 (40 Theory + 24 Lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Deliver a foundational understanding of the core concepts of Machine Learning.
2. Impart the knowledge to apply common supervised learning algorithms to solve classification and regression problems.
3. Demonstrate the techniques essential to uncover hidden patterns and structures within unlabeled data.
4. Make use of Multilayer Perceptron via back-propagation algorithm to solve complex classification problems.
5. Evaluate the performance of machine learning models using appropriate metrics and statistical techniques.

Module 1: Introduction To Machine Learning	**No. of Hrs: 8**

Introduction to Machine Learning Introduction, Examples of Machine Learning Applications: Learning Associations, Classification, Regression, Unsupervised Learning, Reinforcement Learning.

Practical Programs	**No. of Hrs: 4**

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

Module 2: Supervised Learning	**No. of Hrs: 8**

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

Practical Programs	**No. of Hrs: 5**

3. **Decision Tree - Classification:** Classify data points based on a tree-like structure with decision rules.
4. **Support Vector Machine (SVM) - Classification:** Create a hyperplane to separate data points into classes with maximum margin.
5. **Naive Bayes Classification:** Classify data points based on Bayes' theorem and assuming independence of features.

Module 3: Unsupervised Learning	**No. of Hrs: 8**

Clustering: Introduction, Mixture Densities, k-Means Clustering, Supervised Learning after Clustering, Hierarchical Clustering, Nonparametric Methods: Nonparametric Density Estimation, Histogram Estimator, Kernel Estimator, k-Nearest Neighbor Estimator, Decision Trees: Introduction, Univariate Trees, Classification Trees, Regression Trees.

Practical Programs	**No. of Hrs: 5**

6. **K-Nearest Neighbors (KNN)** - Classification: Classify data points based on the majority vote of their K nearest neighbors.
7. **K-Means Clustering**: Group data points into K clusters based on similarity.

8. **Principal Component Analysis (PCA)** - Dimensionality Reduction: Reduce the number of features while retaining most of the information.

Module 4 : Multilayer Perceptrons	**No. of Hrs: 8**

Multilayer Perceptrons : Introduction, Understanding the Brain, Neural Networks as a Paradigm for Parallel Processing, The Perceptron, Training a Perceptron, Learning Boolean Functions, Multilayer Perceptrons, MLP as a Universal Approximator, Backpropagation Algorithm, Nonlinear Regression, Two-Class Discrimination, Multiclass Discrimination, Multiple Hidden Layers.

Practical Programs	No. of Hrs: 5

9. Perceptron: Implement a perceptron and train it to classify the AND, OR, and XOR logic gates.
10. **Parallel processing:** Simulate a simple neural network to understand the concept of parallel processing and information flow.

Module 5 : Reinforcement Learning And Evaluating Hypotheses	**No. of Hrs: 8**

Reinforcement Learning: Elements of Reinforcement Learning, Model-Based Learning, Value Iteration, Policy Iteration, Temporal Difference Learning, Exploration Strategies, Deterministic Rewards and Actions, Nondeterministic Rewards and Actions, Eligibility Traces, Generalization, Design and Analysis of Machine Learning Experiments: Guidelines for Machine Learning Experiments, Cross-Validation and Resampling Methods, K-Fold Cross-Validation, Cross-Validation, Bootstrapping.

Practical Programs	**No. of Hrs: 5**

11. Implement a reinforcement learning agent to navigate a grid world. The agent must find the optimal path from a starting point to a goal state while avoiding obstacles.
12. 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: Identify the appropriate machine learning paradigm for a given problem.
CO2: Utilize supervised learning models to make predictions based on labeled data.
CO3: Select meaningful insights from unlabeled data through clustering and dimensionality reduction techniques.
CO4: Apply Multilayer Perceptron to solve complex classification tasks.
CO5: Identify and compare the performance of various machine learning models.

Textbooks:

1. Ethem Alpaydin, "*Introduction to Machine Learning"*, 3rd Edition, MIT Press, Prentice Hall of India, 2014.
2. Andreas C. Müller and Sarah Guido, "*Introduction to Machine Learning with Python A Guide for Data Scientists"*. 1st Edition O'Reilly Media, Inc.

Reference Books:

1. Tom Mitchell, "*Machine Learning"*, 3rd Edition McGraw Hill, 1997
2. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", Pearson Education, 6th Edition, 2009.
3. Stephen Marsland, "*MACHINE LEARNING - An Algorithmic Perspective"*, 2nd Edition, 2015.

Web Links:

1. https://www.w3schools.com/python/python_ml_getting_started.asp
2. https://www.geeksforgeeks.org/machine-learning/
3. https://www.tutorialspoint.com/machine_learning/index.htm

NATURAL LANGUAGE PROCESSING			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCPE554**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**3:0:2**	Exam Hrs	**03**
Total Hours	**64(40 hrs theory + 24 hrs lab)**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Understand the basic concepts of Natural Language Processing.
2. Illustrate different Language Modeling techniques.
3. Acquire knowledge on various Levels of NLP analysis.
4. Analyze NLP algorithms.

Module 1 : Overview	**No. of Hrs: 13**

Overview and language modeling: Overview: Origins and challenges of NLP-Language and Grammar-Processing Indian Languages- NLP Applications-Information Retrieval. Language Modeling: Various Grammar - based Language Models-Statistical Language Model.

Laboratory Components:

1. Develop a grammar checker that identifies and suggests corrections for grammatical errors in English sentences using rule-based or statistical methods.
2. Build a toolkit that includes basic processing tasks (tokenization, stemming, POS tagging) for
   Indian languages such as Hindi or Bengali.

Module 2 : Word level and syntactic Analysis	**No. of Hrs: 13**

Word level and Syntactic Analysis: Word Level Analysis: Regular Expressions-Finite- State Automata-Morphological Parsing-Spelling Error Detection and correction-Words and Word classes-Part-of Speech Tagging.

Syntactic Analysis: Context-free Grammar- Constituency- Parsing-Probabilistic Parsing.

Laboratory Components:

3. Implement a program that detects and corrects spelling errors in a text corpus using techniques like edit distance and dictionary lookup.

Module 3 : Semanic Analysis and Discourse Processing	**No. of Hrs: 12**

Semantic Analysis : Introduction, Meaning representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation.

Discourse Processing: Introduction Cohesion, Reference Resolution, Discourse Coherence and Structure.

Laboratory Components:

4. Create a system that disambiguates word senses in context using WordNet or other lexical resources and algorithms like Lesk's method.
5. Implement a coreference resolution system that identifies and links pronouns and named entities to their referents across a document.

Module 4 : Natural Language Generation and Machine Translation	**No. of Hrs: 13**

Natural Language Generation: Architectures of NGL Systems, Generation tasks and representations, Applications of NGL

Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Direct Machine Translations, Rule - based Machine Translation, Corpus-based Machine Translation.

Laboratory Components:

6. Develop a machine translation system that translates text between English and a chosen Indian language (e.g., Hindi) using statistical methods or neural networks.

Module 5 : Information Retrieval & Applications	**No. of Hrs: 13**

INFORMATION RETRIEVAL AND Other Applications: Information Retrieval: Design features of Information Retrieval Systems-Classical, Non classical, Alternative Models of Information Retrieval, Evaluation of the IR System, NLP in IR, Relation Matching, Knowledge based Approaches, Conceptual Graphs in IR. Other Applications: Information extraction, Automatic Text Summarization.

Laboratory Components:

7. Design and implement an information retrieval system that allows users to search and retrieve relevant documents using techniques like vector space models and relevance feedback.

Learning Outcomes:

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

CO1: Apply Natural Language Modelling Approaches to a variety of applications.
CO2: Make use of Word level, Syntactic level, Semantic level analysis on a few applications.
CO3: Apply the machine translation approaches to a few problems.
CO4: Identify the different design models of Information Retrieval Systems and use it in certain scenarios.

Textbooks:

1. Tanveer Siddiqui, U.S. Tiwary, "*Natural Language Processing and Information Retrieval"*, Oxford University Press, 2008.
2. Anne Kao and Stephen R. Poteet, "*Natural Language Processing and Text Mining"*, Springer-Verlag London Limited, 2007.

Reference Books:

1. Daniel Jurafsky and James H Martin*, "Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition"*, 2nd Edition, Prentice Hall, 2008.
2. James Allen, "*Natural Language Understanding"*, 2nd Edition, Benjamin/Cummings publishing company, 1995.
3. Gerald J. Kowalski and Mark.T. Maybury, "*Information Storage and Retrieval systems"*, Kluwer academic Publishers, 2000.

Web Links:

1. https://www.youtube.com/watch?v=6P2z9PDRWTw
2. www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/
3. ttps://www.tutorialspoint.com/natural_language_processing/natural_language_processing_synta ctic_analysis

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

MINI PROJECT			
Semester	**II**	CIE Marks	**50**
Course Code	**23MCSE527**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**0:0:4**	Exam Hrs	**2.5**
Total Hours	**26**	Credits	**02**

Course Learning Objectives:

This course is designed to

1. Recall the concepts learnt in Database Management Course
2. Apply the required tools and techniques to develop a software application
3. Examine the requirements and transform them to a software module(s)
4. Formulate the test cases and strategies for the software module(s)

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.

- The marks of the mini-project would be given on the basis of performance in CIE and SEE.

Evaluation:

During mini-project work, the evaluation process will be divided into a number of phases to assess the continuous progress (Minimum three phases, after every phase a review will be conducted). The mini project guides and project coordinator will adopt rubrics, which is set by the Department for evaluation. This would later be submitted to the Head of the Department. Each internal guide will verify the problem statement of the project and the associated literature and the implementation details. The Department will always encourage students to publish their work in standard conferences/journals. The evaluation of the mini project (CIE & SEE) will be based on the rubrics set from time to time.

Review #	Agenda	Assessment	Review Assessment Weightage	Overall Weightage
Review 1 (R1)	Project Synopsis Evaluation	Rubrics 1	15 marks	30 (R1 + R2)
Review 2 (R2)	Mid-Term Project Evaluation	Rubrics 2	15 marks	
Report Preparation (R3)		Rubrics 3	20 marks	20 marks
Final SEE Project Viva-Voce	End-Semester Project Evaluation		50 marks	50 marks
Total				**100 marks**

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	**23MCPC611**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**4:0:0**	Exam Hrs	**03**
Total Hrs	**52**	Credits	**04**

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 Hrs: 12**

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.

Text book 1- Ch 3,4

Module 2: Cloud Infrastructure and Services	**No. of Hrs: 10**

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, **Amazon web services as a cloud platform:** Compute services, Storage Services, Communication services. Additional services, **Case Study**: Understanding services from major providers (Google Cloud, Microsoft Azure)

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

Module 3: Cloud Containers and its Deployment	**No. of Hrs: 10**

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, Load Balancer.

Textbook 4 - Ch 1, 8, 9, 10

Module 4: Cloud Security and Compliance	**No. of Hrs: 10**

Data security and storage: Aspects of data security, Data Security Mitigation, Provider data and its security, **Identity and Access Management (IAM):** Definition, IAM standards and protocols, IAM practices in cloud. **Audit and compliance:** Internal Policy Compliance, Governance, risk and compliance, Illustrative control objectives for Cloud computing.
Textbook 3 - Ch 4, 5, 8

Module 5: Emerging Trends in cloud computing	**No. of Hrs: 10**

Edge and Fog Computing: Edge computing architecture, Fog computing architecture, Use cases of edge and fog computing in IoT. **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: Summarize the foundational principles of cloud computing.
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, IAM standards 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	**23MCPC612**	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. 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 Hrs: 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 Hrs: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 Hrs: 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, The RandomAccess Interface, Working with Maps.

Generics in Java: Generics Example, Implementing Generic classes and methods, Using Wildcard Arguments, Generic Interfaces, Generic Class Hierarchies, 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.
9. Write a program that uses Comparable and Comparator for custom sorting, using a PriorityQueue.

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

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

Textbook 3: Ch 1, 3, 4, 6, 7, 8

Laboratory Components

10. Create a basic Spring Boot project and develop REST APIs.
11. Implement a data access layer using Spring Data JPA to interact with a database. Perform CRUD operations through REST APIs.
12. 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 Hrs: 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

13. Set up Hibernate with MySQL/PostgreSQL and perform basic CRUD operations.
14. Map Java classes to database tables and implement relationships (One-to-Many, Many-to-Many).
15. Write HQL queries to retrieve data, implement filtering, sorting, and joins using HQL.

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

BIG DATA ANALYTICS			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPC613**	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. Impart knowledge on the fundamental concepts and evolution of Big Data, the role of Big Data analytics and Hadoop as key technologies for managing and processing large-scale data.
2. Provide an understanding of the Hadoop Ecosystem, HDFS architecture, and the MapReduce framework.
3. Familiarize with the core fundamentals of the MapReduce framework, its optimization techniques, and explore virtualization technologies that support Big Data environments.
4. Enable students with an understanding of data processing with Apache Pig and Hive, and explore the characteristics and querying capabilities of HBase within Big Data environments.

Module 1 : Introduction to Big Data	**No. of Hrs: 08+04**

Big Data Overview: Characteristics of Big Data (Volume, Variety, Velocity, Veracity, and Value). Big Data Storage and Analysis, Comparison with other systems, A brief history of Hadoop.

Textbook 1 - Ch 1

Textbook 2 - Ch 1

Laboratory Components

1. Use a large dataset (e.g., social media data, sensor data) to perform the following:
   Volume: Calculate the size of the dataset and visualize its growth over time.
   Variety: Identify different data types (structured, semi-structured, unstructured) present in the dataset.
   Velocity: Analyze the rate of incoming data over a defined period.
   Veracity: Assess the quality of the data, including missing values and inconsistencies.
   Value: Derive insights or patterns from the dataset that provide business value.
2. Install Hadoop in a local or cloud environment. Perform the following HDFS operations:
   - Create a directory in HDFS.
   - Upload files to HDFS.
   - List files and directories in HDFS.
   - Download files from HDFS to the local filesystem.
   - Delete files from HDFS

Module 2: Hadoop Ecosystem & HDFS	**No. of Hrs: 08+04**

Hadoop Ecosystem: Overview of tools and techniques.
HDFS (Hadoop Distributed File System): Design of HDFS, HDFS Concepts: blocks, Namenodes and Datanodes, HDFS Federation. HDFS Filesystem. Reading data from a Hadoop URL, Reading data using the FileSystem API.
Textbook 1 - Ch 4
Textbook 2 - Ch 3
Laboratory Components

3. Develop a program to upload a file to HDFS, create a directory, and manage files in HDFS.
4. Implement a program to read a file from HDFS and download it to the local file system.
5. Implement a program to delete a file from HDFS and check disk usage in HDFS.

Module 3: MapReduce, Big data technology	**No. of Hrs: 08+06**

Understanding Map Reduce Fundamentals: The Map Reduce Framework, Exploring the Features of Map Reduce, Working of Map Reduce, Exploring Map and Reduce Functions, Techniques to Optimize Map Reduce Jobs. Exploring big data stack.
Virtualization and big data: Virtualization environment, Virtualization approach.
Data and storage virtualization: Managing virtualization with hypervisor, architecture of hypervisor technology.
Textbook 1 - Ch 5, 6
Laboratory Components

6. Develop a MapReduce program to calculate the frequency of a given word in a given file.
7. Develop a MapReduce program to find the maximum temperature in each year.
8. Develop a MapReduce program to find the grades of a student.
9. Develop a MapReduce to find the maximum electrical consumption in each year given electrical consumption for each month in each year.

Module 4 : YARN and Hive	**No. of Hrs: 08+06**

YARN: Architecture, working of YARN, YARN schedulers, configuration and commands.
Introduction to Hive: Hive services, Data types in Hive, Built-in functions, Hive DDL, Data manipulation in Hive, Data Retrieval Queries in Hive.
Textbook 1 - Ch 11, 12
Laboratory Components

10. Implement a program to:
    - Create a Hive table using various data types.
    - Insert data into the created Hive table.
11. Develop a program to utilize Hive built-in functions to perform operations on data.
12. Develop a program to retrieve specific data from the Hive table using SELECT queries.
13. Demonstrate how to drop a Hive table.

Module 5 : Data Analytics using Pig	**No. of Hrs: 08+04**

Introduction to Pig, Pig architecture, Pig modes.
Pig Latin: Structure, Statements, Expressions, Types, Schemas, Operators, Macros. User-Defined functions, Data processing operators.

Textbook 1 - Ch 13
Textbook 2 - Ch 11

Laboratory Components

14. Implement a program to create a Hive table for sales data and then execute a Pig query to calculate total sales for each product.
15. Implement a Pig program to execute an interactive query to filter sales data based on given conditions.
16. Implement a Pig program to calculate average scores from a dataset.
17. Implement a Pig program to retrieve the top scorers from a dataset.

Course Outcomes:

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

CO1: Outline the evolution of Big Data, significance of Big Data analytics, and the functionalities of Hadoop.
CO2: Summarize the different components of the Hadoop ecosystem, the working of the MapReduce framework, and the role of HBase in storing and processing large datasets.
CO3: Apply optimization techniques to enhance job performance, and evaluate the impact of virtualization technologies on Big Data processing environments.
CO4: Utilize Hive functionalities, and perform querying operations in HBase.
CO5: Build data process scripts using Apache Pig.

Textbooks:

1. DT Editorial Services, *"BIG DATA, Black Book™"*, DreamTech Press, 2020.
2. Tom White, "*Hadoop: The Definite Guide"*, 3rd Edition. O'Reilly, 2012.

Reference Books:

1. Balamurugan Balusamy, Nandhini Abirami R, Seifedine Kadry and Amir Gandomi, "*Big Data: Concepts, Technology and Architecture"*, Wiley, 2023.
2. Michele Chambers, Andre Tantillo, and Sangeet Chaudhary, "*Big Data: Principles and Practices"*, McGraw-Hill Education, 2014.

Web Links:

1. Big data hadoop tutorial for beginners: https://hadoop.apache.org/
2. Hadoop and MapReduce: https://www.databricks.com/glossary/mapreduce
3. Apache Pig: https://pig.apache.org/
4. Big data Hadoop course: https://www.youtube.com/watch?v=1vbXmCrkT3Y

OBJECT ORIENTED MODELING AND DESIGN			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPC614**	SEE Marks	**50**
Teaching Hrs/Week (L: T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**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 Hrs: 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

Laboratory Components

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: State Modeling and Advanced State Modeling	**No. of Hrs: 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

Laboratory Components

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 ArgoUML tool to represent the different states while booking an airline ticket. Show the events that trigger the transitions.

Module 3: Interaction Modeling and Advanced Interaction Modeling	**No. of Hrs: 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

Laboratory Components

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 Hrs: 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

Laboratory Components

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 Hrs: 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 V, "*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. Craig Larman, "*Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design"*, 2nd Edition, Prentice Hall Professional, 2002.
3. Brett McLaughlin, Gary Pollice, and David West, "*Head First Object-Oriented Analysis and Design*", 1st Edition, O'Reilly Media, Inc., 2006.
4. Grady Booch, James Rumbaugh, and Ivar Jacobson, "*The Unified Modeling Language User Guide",* 2nd Edition, Addison-Wesley, 2005.

Web Links:

1. UML Diagrams - An 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/

ETHICAL HACKING			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPE651**	SEE Marks	**50**
Teaching Hrs/Week (L: T: P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Instill the basics of penetration tools and methodologies used in ethical hacking.
2. Impart knowledge in analyzing the vulnerabilities and attacks on a system.
3. Provide knowledge on exploitation and attacks.
4. Familiarize with the process of phishing attacks.
5. Impart knowledge on web application vulnerabilities, focusing on SQL injection.

Module 1: Introduction to Penetration Testing	**No. of Hrs: 06+04**

Stages of Penetration Test: Pre-engagement, Information gathering, Threat modeling, Vulnerability analysis, Exploitation, Post exploitation, Reporting, Kali Linux: User privileges, Data manipulation, Managing networking, Metasploit framework: Finding Metasploit modules, Setting module options, Payloads, Types of Shells, Setting a payload manually.

Text book 1: Ch 1, 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 2: Information Gathering and Vulnerabilities	**No. of Hrs: 05+04**

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 Scanner Modules.

Text book 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 3: Exploitation and Attacks	**No. of Hrs: 05+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.

Text book 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 4: Social Engineering	**No. of Hrs: 05+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, A Sophisticated and Sneaky Social Engineering Attack, Faking Emails: Performing a DNS Lookup of a Mail Server, Communicating with SMTP, Writing an Email Spoofer, Spoofing SMTPS Emails, Faking Websites.

Text book 1: Ch 11, T2: Ch 7

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.

Module 5: Web Application Testing	**No. of Hrs: 05+04**

SQL Injection, XPath Injection, Local File Inclusion, Remote File Inclusion, Command Execution, Cross-Site Scripting, Cross-Site Request Forgery, Web Application Scanning with w3af.

Text book 1: Ch 14

Laboratory Components:

9. Test a web application for SQL injection vulnerabilities and propose mitigation strategies.
10. Conduct a cross-site scripting (XSS) attack to demonstrate how malicious code can be injected.

Course Outcomes:

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

CO1: Apply penetration testing techniques to identify vulnerabilities and weaknesses in systems.

CO2: Implement basic scripting to connect to a port and scan the network and host.

CO3: Demonstrate the process of payload injection and utilization.

CO4: Identify the process of phishing attacks and assess the security levels involved given a scenario.

CO5: Execute tests to identify and evaluate web application vulnerabilities, focusing 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.

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

DIGITAL FORENSICS			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPE652**	SEE Marks	**50**
Teaching Hrs/Week (L: T:P)	**3:0:0**	Exam Hrs	**03**
Total Hrs	**40**	Credits	**04**

Course Learning Objectives:

This course is designed to

1. Impart knowledge on computer forensics, including its principles, methodologies, and role in modern investigative processes.
2. Familiarize in conducting systematic investigations and executing detailed forensic analysis.
3. Provide the ethical and legal principles required for conducting investigations related to cybercrimes.
4. Instill the ability to critically review and analyze digital evidence, fostering the development of effective investigative strategies.
5. Provide knowledge to apply ethical and legal principles in conducting email investigations, ensuring compliance with privacy regulations and chain of custody requirements.

Module 1: Computer forensics fundamentals	**No. of Hrs: 08**

An Overview of Digital Forensics, Preparing for Digital Investigations, Maintaining Professional Conduct, Preparing a Digital Forensics Investigation, Procedures for Private-Sector High-Tech Investigations, Understanding Data Recovery Workstations and Software, Conducting an Investigation.

Textbook 1: Ch 1

Module 2: Data Acquisition	**No. of Hrs: 08**

Understanding Storage Formats for Digital Evidence, Determining the Best Acquisition Method, Using Acquisition Tools, Validating Data Acquisitions, Performing Raid Data Acquisitions, Using Remote Network Acquisition Tools, Using Other Forensics Acquisition Tools.

Textbook 1: Ch 3

Module 3: Processing Crimes and Incident Scenes	**No. of Hrs: 08**

Identifying Digital Evidence, Collecting evidence in Private-Sector Incident Scenes, Processing Law Enforcement Crime Scenes, Preparing for a Search, Securing a Digital Incident or Crime Scene, Seizing Digital Evidence at the Scene, Storing Digital Evidence, Obtaining a Digital Hash, Reviewing a Case.

Textbook 1: Ch 4

Module 4: Current Digital Forensics Tools	**No. of Hrs: 08**

Evaluating Digital Forensics Tool Needs, Digital Forensics Software Tools, Digital Forensics Hardware Tools, Validating and Testing Forensics Software, E-Mail Investigations: Investigating Email Crime and Violations, Understanding E-Mail Servers, Using Specialized E-Mail Forensics Tool, Applying Digital Forensics Methods to Social Media Communications.

Textbook 1: Ch 6, Ch 11

Module 5: Digital Forensics Analysis, Validation and Virtual Machine Forensics	**No. of Hrs: 8**

Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, An Overview of Virtual Machine Forensics, Performing Live Acquisitions, Network Forensics Overview.
Textbook 1: Ch 9, Ch 10

Course Outcomes:

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

CO1: Identify the fundamental concepts in computer forensics and its associated systems.

CO2: Apply systematic approaches to conduct the systematic investigations and executing detailed Forensic analyses.

CO3: Use the different data acquisition methods and choose an appropriate approach for a given case.

CO4: Solve a forensics case with the available digital evidence and using the appropriate strategies.

CO5: Apply ethical and legal principles in conducting email investigations and ensuring compliance with privacy regulations

Textbooks:

1. Bill Nelson, Amelia Phillips, Chris Stuart, "*Guide to Computer Forensics and Investigations"*, Thomson Course Technology, 6th Edition, 2018.

Reference Books:

1. John R Vacca, Computer Forensics, "*Computer Crime Scene Investigation",* Charles River Media, 2nd Edition, 2005.
2. John Sammons, "*The basics of digital forensics: The primer for getting started in digital forensics",* Elsevier Science, 2014.
3. Linda Volonino, Reynaldo Anzaldua, and Jana Godwin, "*Computer Forensics: Principles and Practices"*, Pearson, 2007.

Web Links:

1. Computer Forensics Specialization : https://www.coursera.org/specializations/computerforensics
2. Digital Forensics and Electronic Evidence: https://www.udemy.com/course/digital-forensics-and-electronic-evidence
3. Cyber Forensics : https://www.mygreatlearning.com/academy/learn-for-free/courses/cyber-forensics

DEEP LEARNING			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPE653**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Impart knowledge on the mathematical, statistical and computational challenges of building neural networks.
2. Provide computational knowledge on deep neural networks.
3. Demonstrate the applications of CNN in various solutions.
4. Familiarize the uses and applications of Auto encoders in different real time problems.
5. Impart deep learning skills and techniques to develop and optimize real-time applications.

Module 1 : Foundation of Neural Networks and Deep Learning	**No. of Hrs: 06+04**

Mathematical foundation in Neural networks. Neural Networks: The Biological Neuron, The Perceptron, Multilayer Feed-Forward Networks. Training Neural Networks: Backpropagation Learning. Activation Functions, Loss Functions. Hyperparameters: Learning Rate, Momentum. Introduction to Deep Learning.

Text book 1 - Ch 1, 2

Laboratory component

1. Implement a simple feed-forward neural network (using NumPy) that can classify a dataset (e.g., the Iris dataset or a simple binary classification dataset).
2. Create a neural network that utilizes different activation functions and evaluate their performance on a classification task.

Module 2: Deep Neural Networks	**No. of Hrs: 05+06**

Common Architectural Principles of Deep Networks: Parameters, Layers, Activation functions, Loss functions, Optimization methods. Hyperparameters: Magnitude (momentum, learning rate), Regularization (dropout, drop connect, L1, L2), Weight initialization strategy, Settings for epochs during training (mini-batch size), Normalization scheme for input data, Building Blocks of Deep Networks.

Text book 1 - Ch 3

Laboratory component

3. Implement a deep feedforward neural network using a framework incorporating various regularization techniques to evaluate their effects on model performance.
4. Implement a program to investigate the effect of different optimization techniques on the training of a deep neural network for a regression or classification task.

Module 3: Convolutional Neural Networks	**No. of Hrs: 05+04**

Unsupervised Pretrained Networks: Autoencoders, Deep Belief Networks (DBNs).
Convolutional Neural Networks (CNNs), Biological Inspiration, Intuition, CNN Architecture Overview, Input Layers, Convolutional Layers, Pooling Layers, Fully Connected Layers, Other Applications of CNNs.

Text book 1 – Ch 4

Laboratory component

5. Develop a CNN to classify images from a dataset and evaluate its performance compared to a basic feed-forward neural network.

Module 4 : Recurrent Neural Networks	**No. of Hrs: 05+04**

Modeling the Time Dimension, 3D Volumetric Input, Markov Models, General Recurrent Neural Network Architecture, LSTM Networks, Domain-Specific Applications and Blended Networks, Network Architecture, Varieties of Recursive Neural Networks, Applications of Recursive Neural Networks.

Text book 1 – Ch 4

Laboratory component

6. Create an RNN to perform a sequence prediction task, such as predicting the next character in a sequence or generating text based on a seed input.

Module 5 : Deep Learning	**No. of Hrs: 05+06**

Linear Factor Models: Probabilistic PCA and Factor Analysis, Independent Component Analysis (ICA).

Autoencoders: Undercomplete Autoencoders, Regularized Autoencoders, Stochastic Encoders and Decoders, Denoising Autoencoders.

Representation Learning: Greedy Layer-Wise Unsupervised Pre-Training, Transfer Learning and Domain Adaptation.

Text book 2 – Ch 18

Laboratory component

7. Develop a program that implements Probabilistic Principal Component Analysis (PPCA) and Factor Analysis to analyze a dataset and visualize the results.
8. Create and evaluate different types of autoencoders, including under complete, regularized, and denoising autoencoders, to understand their functionalities and applications in representation learning.

Course Outcomes:

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

CO1: Outline the basics of mathematical foundation to build Deep Learning models.
CO2: Apply the concepts of deep networks in deep learning.
CO3: Apply the concepts of Convolutional Neural Networks to perform deep learning.
CO4: Utilize deep learning architectures for processing text & image data.
CO5: Make use of various deep learning techniques to design efficient algorithms for real-world applications.

Textbooks:

1. Josh Patterson and Adam Gibson, "*Deep learning A Practitioner's Approach",* 1st Edition, O'Reilly, 2017.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "*Deep Learning"*, 1st Edition, An MIT Press book, 2016.

Reference Books:

1. Deng & Yu, "*Deep Learning: Methods and Applications"*, 1st Edition, Now Publishers, 2013.
2. Douwe Osinga, "*CookBook Deep Learning"*, 1st Edition, O'Reilly, 2017.

Web Links:

1. MATLAB for DeepLearning: https://tinyurl.com/matlabfordl
2. Neural Networks and Deep Learning: https://tinyurl.com/neuralnetworkanddeeplearning
3. Introduction to Deep Learning: https://tinyurl.com/introdcutiontodeeplearning
4. Deep Learning Course: https://www.youtube.com/watch?v=lhufOy2W3Ps

PATTERN RECOGNITION			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPE654**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**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 Hrs: 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:
   a. Load a dataset (e.g., the Titanic dataset or any relevant dataset).
   b. Define classification rules based on specific features (e.g., age, gender, class) and implement a simple rule-based classifier.
   c. Create a function that applies these rules to classify instances in the dataset (e.g., predicting survival based on rules).
   d. Test the classifier on a subset of data and report the classification results.
   e. 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 Hrs: 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 Hrs: 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 Hrs: 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.
   e. 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 Hrs: 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.
   c. 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.
   f. 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/

MAJOR PROJECT PHASE - 1			
Semester	**III**	CIE Marks	**100**
Course Code	**23MCSE615**	SEE Marks	**-**
Teaching Hours/Week (L:T:P)	**-**	Exam Hrs	**03**
Total Hours	**-**	Credits	**05**

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 project report.

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 industries to finalize the topic of the major project.
- Developing a Software Requirements Specification (SRS) document.
- Preparation of synopsis
- Seminar Presentation

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.

INTERNSHIP			
Semester	**III**	CIE Marks	**100**
Course Code	**23MCSE616**	SEE Marks	**-**
Teaching Hours/Week (L: T:P)	**-**	Exam Hrs	**-**
Total Hours	**-**	Credits	**06**

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 10 weeks. This is a mandatory internship that should be completed by a student during the intervening vacation of second and third 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 before the internship starts and this information will be disseminated to the students before their internship starts in the industry.
- 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.

IV SEM

SECURE SOFTWARE DEVELOPMENT			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE661**	SEE Marks	**50**
Teaching Hrs/Week (L: T:P)	**3:0:0**	Exam Hrs	**03**
Total Hrs	**42**	Credits	**03**

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. Impart knowledge to prioritize security requirements during the software engineering process.
3. Emphasize the need to apply architectural security practices and risk analysis to design secure software systems.
4. Build proficiency in secure coding practices and security testing throughout the software development lifecycle.
5. Impart skills to apply governance and security management principles in ensuring secure software development within enterprises.

Module 1: Security as a Software Issue	**No. of Hrs: 10**

Introduction, Software Assurance and Software Security, Threats to Software Security, Sources of Software Insecurity, Benefits of detecting software security, Managing secure software development, What makes software secure: Defining properties of secure software, Influencing the security properties of software, Asserting and specifying desired security properties.

Textbook 1: Ch 1, 2

Module 2: Requirements Engineering for Secure Software	**No. of Hrs: 09**

Introduction, Importance of Requirements Engineering for secure software, Quality Requirements, Security Requirements Engineering. Misuse and Abuse Cases: Creating useful misuse cases, An abuse case example, The SQUARE process model: Description, Tools, Expected results, Output from SQUARE process model, Requirements Elicitation, Requirements Prioritization.

Textbook 1: Ch 3

Module 3: Secure Software Architecture and Design	**No. of Hrs: 08**

Introduction, Software Security Practices for Architecture and Design, Architectural Risk Analysis, Software Security Knowledge for Architecture and Design: Security Principles, Security Guidelines, Attack patterns.

Textbook 1: Ch 4

Module 4: Secure Coding, Testing, Security and Software Complexity	**No. of Hrs: 08**

Introduction, 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, System complexity drivers and security.

Textbook 1: Ch 5, 6

Module 5: Governance and Managing Secure Software	**No. of Hrs: 07**

Introduction, Governance and Security, Adopting an Enterprise Software Security framework, How much security is enough, Security and Project management, Maturity of Practice.

Textbook 1: Ch 7

Course Outcomes:

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

CO1: Apply the principles of software assurance and software security and effectively manage secure software development practices.

CO2: Use standard practices for secure requirements engineering and identify misuse or abuse cases for a given scenario.

CO3: Apply secure architectural principles and attack patterns to enhance the security of software designs.

CO4: Use secure coding practices to build reliable and secure software applications.

CO5: Apply security governance principles into software project management and evaluate the maturity of security practices to determine the appropriate level of security.

Textbooks:

1. 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. Michael Howard, Steve Lipner, *The Security Development Lifecycle: SDL, A Process for Developing Demonstrably More Secure Software*, Microsoft Press, 2006,
2. Jason Grembi, *Secure Software Development: A Security Programmer's Guide*, Cengage Learning, 2020.
3. Jan Jürjens, Flavio De Paoli, and Alexander Pretschner, *Engineering Secure Software and Systems*, Springer, 2020.

WEBLINKS

1. The Cyber Security Body of Knowledge:
   https://www.cybok.org/media/downloads/CyBOK_v1.1.0.pdf:
2. Secure Software Design:
   https://www.udemy.com/course/secure-software-design-secure-software-series-course-3-of-8/?couponCode=LETSLEARNNOWPP
3. Secure System Analysis and Design:
   https://www.futurelearn.com/courses/system-analysis-and-design-sc

BLOCKCHAIN TECHNOLOGY			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE662**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 Theory + 24 Lab)**	Credits	**03**

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. Provide knowledge on Bitcoin's key management and transaction mechanisms.
4. Instill the knowledge on development and deployment of smart contracts on the Ethereum blockchain.
5. Provide knowledge on the applications of blockchain's role across different technologies.

Module 1: Fundamentals of Blockchain — **No. of Hrs: 05+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 Hrs: 05+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 and Transaction Processing — **No. of Hrs: 04+06**

Private and public keys in Bitcoin, Typical Bitcoin addresses, Transactions, Bitcoin payments, Bitcoin in practice, Alternative Coins, Bitcoin limitations.

Textbook 1: Ch 6, 7

Laboratory Components

5. Using Bitcoin's public and private key infrastructure, simulate a basic transaction between two parties.
6. 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).
   - Verify each transaction in the chain

Module 4 : Smart Contracts and Ethereum	**No. of Hrs: 05+06**

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 8, 9

Laboratory Components

7. Develop and deploy a simple smart contract using Solidity on the Ethereum test network.
8. Explore Ethereum 101 by creating an Ethereum wallet and executing a transaction on the test network.

Module 5: Blockchain and Allied Technology	**No. of Hrs: 07**

Block chain and Cloud Computing, Characteristics of Blockchain Cloud, Blockchain and Artificial Intelligence, Block chain and IoT, Blockchain and Machine Learning, Blockchain and Robotic Process Automation.

Textbook 2: Ch 11

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: Identify the payment system in Bitcoin using public and private key encryption.

CO4: Utilize the smart contract components and apply knowledge to develop and deploy contracts on the Ethereum blockchain.

CO5: Outline the applications of blockchain across different technologies.

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. Kumar Saurabh, Ashutosh Saxena, "*Blockchain Technology Concepts and Applications"*, Wiley India Pvt. Ltd, 1st Edition, 2023.

Reference Books:

1. Mayukh Mukhopadhyay, *"Ethereum Smart Contract Development: Build Blockchain-Based Decentralized Applications Using Solidity",* Packt, 2018.
2. Roberto Infante, "*Building Ethereum DApps: Decentralized Applications on the Ethereum Blockchain",* Manning Publications, 2019.

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/

COMPUTER VISION			
Semester	**III**	CIE Marks	**50**
Course Code	**23MCPE663**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed for students to:

1. Impart knowledge on the fundamental concepts of image processing.
2. Familiarize with different techniques while enhancing images.
3. Provide knowledge on image segmentation and compression algorithms for image processing.
4. Impart the principles of image formation, image processing algorithms and different algorithms for recognition from single or multiple images.
5. Acquaint the core vision tasks of scene understanding and recognition in real world scenarios.

Module 1 : Introduction to Computer Vision	**No. of Hrs: 06+04**

Introduction to Computer vision, Image formation: Geometric primitives and transformations: Geometric primitives, 2D, 3D transformation, 3D rotation, 3D to 2D projections, Lens distortions, Photometric image formation: Lighting, Reflectance and shading, Optics. The digital camera: Sampling and aliasing, Color, Compression.

Text book 1: Ch 1, 2

Laboratory component

1. Implement basic image transformations and perform perspective correction using a real-world image.
   Tasks:
   a. Load an image (e.g., a photograph of a document taken at an angle).
   b. Apply basic transformations such as scaling, rotation, and translation to the image.
   c. Use the concept of Perspective Transformation to correct the skew in the document image and obtain a properly aligned, front-facing version of the document.
   d. Visualize the original and transformed images.

2. Apply different convolution masks (kernels) to an image to observe the effects of smoothing, sharpening, and edge detection.
   Tasks:
   a. Load an image.
   b. Apply the following convolution masks to the image:
      i. Smoothing (blurring): Use a Gaussian kernel.
      ii. Sharpening: Apply a high-pass filter (e.g., Laplacian mask).
      iii. Edge Detection: Use the Sobel or Prewitt operators to detect edges in both horizontal and vertical directions.

Module 2: Image processing	**No. of Hrs: 05+04**

Point operators: Pixel transforms, Color transforms, Compositing and matting, Histogram equalization. Linear filtering: Separable filtering, Examples of linear filtering, Band-pass and steerable filters. More neighborhood operators, Fourier transforms, Pyramids and wavelets, Geometric transformations, Global optimization.

Text book 1: Ch 3

Laboratory component

3. Write a program to perform the following tasks:
   a. Load a grayscale image and introduce synthetic noise (e.g., Gaussian noise or Salt-and-Pepper noise) to simulate a noisy image.
   b. Display the original and noisy images side by side.
   c. Implement spatial filtering techniques to reduce the noise:
      - Apply a filter to the noisy image.
   d. Display the filtered images alongside the original and noisy images for comparison.

Module 3: Feature Extraction and Image Segmentation	**No. of Hrs: 05+06**

Points and patches: Feature detectors, Feature descriptors, Feature matching, Feature tracking. Edges: Edge detection, Edge linking, Lines: Successive approximation, Hough transforms, Vanishing points.

Text book 1: Ch 4, 5

Laboratory component

4. Implement one or more descriptors and compare their performance.
5. Implement a Hough transform for finding lines in images.
6. Compute the vanishing points in an image using one of the techniques.

Module 4 : Structure and Motion Analysis	**No. of Hrs: 05+04**

Triangulation: Two-frame structure from motion, Factorization, Bundle adjustment, Constrained structure and motion.

Dense motion estimation: Translational alignment, Parametric motion, Spline-based motion, Layered motion.

Text book 1: Ch 7, 8

Laboratory component

7. Write a program to segment an image into separately moving regions or to reliably find motion boundaries.
8. Decompose into separate layers a video sequence of a scene taken with a moving camera.

Module 5 : Recognition	**No. of Hrs: 05+06**

Object detection: Face detection, Pedestrian detection, Face recognition: Eigenfaces, Active appearance and 3D shape models, Instance recognition: Geometric alignment, Large databases. Category recognition: Bag of words, Part-based models, Recognition with segmentation, Context and scene understanding, Recognition databases and test sets: Learning and large image collections.

Text book 1: Ch 9

Laboratory component

9. Build and test one of the face detectors.
    a. Download one or more of the labeled face detection databases
    b. Generate your own negative examples by finding photographs that do not contain any people.
    c. Implement face detectors.
10. For a set of facial photographs build a recognition system to re-recognize a person.
    a. Take several photos of each of your classmates and store them.
    b. Align the images by automatically or manually detecting the corners of the eyes and using a similarity transform to stretch and rotate each image to a canonical position.
    c. Compute the average image and a PCA subspace for the given face images
    d. Take a new set of photographs a week later and use them as your test set.

Course Outcomes:

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

CO1: Identify the different image acquisition and sampling techniques for image formation and transformation.
CO2: Apply different filtering techniques and image compression methods on given problems.
CO3: Utilize image formation and image processing techniques on image detection.
CO4: Apply matching algorithms for various image modalities to implement robust feature detection.
CO5: Make use of computer vision techniques to develop real-world applications across several domains.

Textbooks:

1. Richard Szeliski, "*Computer Vision: Algorithms and Applications",* 2nd Edition, Springer, 2020.

Reference Books:

1. D. Forsyth and J. Ponce, "*Computer Vision – A modern approach"*, 2nd Edition, Prentice Hall, 2012.
2. S. Sridhar, "*Digital Image Processing"*, 2nd Edition, Oxford University Press, 2016.
3. D. H. Ballard, C. M. Brown. "*Computer Vision"*. Prentice-Hall, Englewood Cliffs, 1982.
4. S. Jayaraman, S. Esakkirajan, T. Veerakumar, "*Digital Image Processing"*, Tata McGraw Hill 2014.

Web Links:

1. Computer Vision: https: //www.youtube.com/watch?v=WHgbp6dsk2M
2. Computer Vision with Tensorflow and Keras: https://www.kaggle.com/learn/computer-vision
3. Object detection: https://www.youtube.com/watch?v=WgPbbWmnXJ8

APPLICATIONS OF MACHINE LEARNING FOR IMAGE AND VIDEO ANALYTICS			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE664**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to:

1. Impart the fundamental knowledge on image processing.
2. Instill skills to perform image enhancement and apply filters related to special and frequency domain.
3. Provide knowledge on image segmentation for image analysis.
4. Familiarize students with the general architecture of a face recognition system.
5. Impart the knowledge on video segmentation and shot detection techniques.

Module 1 : Fundamentals of Image processing	**No. of Hrs: 06 + 04**

Introduction – Steps in image processing systems – Image acquisition -Sampling and Quantization – Pixel relationships – Color fundamentals and models, File formats, Image operations – Arithmetic and Morphological.

Textbook 1 - Ch 1, 2

Laboratory components

1. Implement a program to load a grayscale image and apply different levels of quantization (e.g., 2-bit, 4-bit, 8-bit). Display the original image alongside the quantized versions to observe the effect of reduced bit depth on image quality.
2. Implement a program to demonstrate different interpolation techniques that affect image quality, especially along edges, when rotating an image.

Module 2: Image Enhancement	**No. of Hrs: 05 + 04**

Spatial Domain: Gray level Transformations – Histogram processing – Spatial filtering smoothing and sharpening. Frequency Domain: Filtering in frequency domain – DFT, FFT, DCT – Smoothing and sharpening filters – Homomorphic Filtering.

Textbook 1 - Ch 3, 4, 8

Laboratory components

3. Implement a program to load a grayscale image and perform gray level transformations, including contrast stretching, negative transformation, and gamma correction. Display the original and transformed images along with their histograms.
4. Load a grayscale image and convert it to the frequency domain using the Discrete Fourier Transform (DFT). Implement low-pass and high-pass filters in the frequency domain and apply them to observe the effects of frequency-based smoothing and sharpening.

Module 3: Image Segmentation and Feature Analysis	**No. of Hrs: 05 + 04**

Detection of Discontinuities – Edge operators - Edge linking and Boundary Detection - Thresholding - Region based segmentation – Morphological Watersheds – Motion Segmentation.

Textbook 1 - Ch 9, 10

Laboratory components

5. Implement a program to segment an image using thresholding methods. Then, apply region-based segmentation techniques, to extract distinct regions in the image.
6. Use morphological operations to preprocess a video sequence and apply the watershed algorithm to segment moving objects.

Module 4 : Image Analysis	**No. of Hrs: 05 + 06**

Introduction, Image Recognition: General Approach, Image Detection and Localization: Image Segmentation and Normalization with TorchVision, Lighting Normalization: Center/Surround Retinex, Gross and Brajovic's Algorithm, Normalization with TorchVision, Feature Extraction: Holistic Approaches, Local Approaches, Feature Extraction with TorchVision, Classification, Performance Assessment.

Textbook 2 - Ch 13

Laboratory components

7. Implement face detection and localization using pre-trained models available in TorchVision.
8. Extract features from facial images using holistic and local approaches with TorchVision.
9. Classify faces using extracted features and evaluate the classification performance.

Module 5: Video Analysis	**No. of Hrs: 05 + 06**

Introduction, Applications of Video Segmentation, Shot Boundary Detection: Pixel-Based Approaches, Block-Based Approaches, Histogram-Based Approaches, Clustering-Based Approaches, Performance Measures, Shot Boundary Detection with Torchvision, Keyframe Extraction, Keyframe Extraction with Torchvision and Torch.

Textbook 2 - Ch 14

Laboratory components

10. Implement shot boundary detection using pixel-based approaches and block-based approaches for a given video.
11. Develop a program to extract keyframes from a video using histogram-based approaches.
12. Apply Torch and Torchvision to extract keyframes based on clustering-based approaches.

Course Outcomes:

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

CO1: Outline the fundamentals of image processing.
CO2: Apply gray-level transformations and implement special and frequency domain filtering.
CO3: Make use of various approaches to achieve improvement in segmentation performance.
CO4: Apply different types of face recognition mechanisms to solve real-world problems.
CO5: Utilize various video segmentation techniques, and assess their performance.

Textbooks:

1. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 3rd Edition, Pearson Education, 2009.
2. Francesco Camastra, Alessandro Vinciarelli, "*Machine Learning for Audio, Image and Video Analysis",* SPIN Springer's internal project number, 2007.

Reference Books:

1. Milan Sonka, Vaclav Hlavac, Roger Boyle, "*Image Processing, Analysis, and Machine Vision"*, 4th Edition, Thomson Learning, 2013.
2. Caifeng Shan, FatihPorikli, Tao Xiang, Shaogang Gong, "*Video Analytics for Business Intelligence"*, Springer, 2012.

Web Links:

1. Real time analytics for Image and Video: https://tinyurl.com/mlimagevideo
2. Real Time Video Analytics: https://www.youtube.com/watch?v=uSgwTQV45IY
3. Image recognition: https://cloud.google.com/vision/
4. Video data processing**:** https://www.youtube.com/watch?v=AxIc-vGaHQ0

DISASTER MANAGEMENT AND BUSINESS CONTINUITY IN CYBERSECURITY			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE671**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**3:0:0**	Exam Hrs	**03**
Total Hrs	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to:

1. Impart the fundamental concepts of disaster and risk management frameworks.
2. Enable students to develop and manage incident response strategies for cyber incidents.
3. Familiarize students with the different risk mitigation strategies and its impact on a business.
4. Instill the ability to create and use disaster recovery and business continuity plans.
5. Provide the skills to respond during an emergency, testing and auditing a disaster recovery plan.

Module 1: Introduction to Disaster Management	**No. of Hrs: 08**

Definition, Scope of cyber security disasters, Overview of disaster management frameworks, Major cyber security incidents (Case Studies : SolarWinds, WannaCry, NotPetya), Identifying and assessing cyber security risks, Key concepts: Risk, Vulnerability, threat, impact, Risk management frameworks : NIST, ISO27001.

Textbook 3 - Part 1

Module 2: Incident Response Frameworks	**No. of Hrs: 08**

Significance, Strategy Vs Tactics, Prerequisites : Identify and Protect functions, Defined Cybersecurity program, Support for Incident Response, Incident Response frameworks : NIST 800-61, Program Implementation Guidance.

Textbook 2 - Ch 1, 2, 3

Module 3: Business Impact Analysis & Risk Mitigation Strategy	**No. of Hrs: 08**

Business Impact Analysis Overview, Understanding Impact Criticality, Identifying Business Functions, Gathering Impact Data, Determining Impact, BIA Data Points, Industry Spotlight #2.

Types of Risk Mitigation Strategies, Risk Mitigation Process, IT Risk Mitigation, Backup and Recovery Considerations.

Textbook 1 - Ch 5, 6

Module 4: Business Continuity and Disaster Recovery Plan Development	**No. of Hrs: 08**

Phases of Business Continuity and Disaster Recovery, Defining BC/DR Teams and Key Personnel, Defining Tasks and Assigning Resources, Communications Plans, Event Logs and Change Control, Industry Spotlight #3.

Emergency Response and Recovery: Emergency Management Overview, Emergency Response Plans, Crisis Management Team, Disaster Recovery, IT Recovery, Business Continuity, Industry Spotlight #4.

Textbook 1 - Ch 7

Module 5: Training, Testing, Auditing & Maintenance	**No. of Hrs: 10**

Training for Emergency Response, Disaster Recovery, and Business Continuity, Testing your Business Continuity and Disaster Recovery Plan, Performing IT Systems Audits.

BC/DR Plan Change Management, Strategies for Managing Change, BC/DR Plan Audit, Plan Maintenance Activities, Project Close Out, Checklists.

Textbook 1 - Ch 9, 10

Course Outcomes:

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

CO1: Identify various cyber threats and apply principles of disaster management to mitigate risks effectively.

CO2: Construct an incident response plan using an incident response framework for a given threat scenario.

CO3: Develop risk mitigation strategies for a typical business function.

CO4: Identify the roles and responsibilities of a BC/DR team, including the Crisis Management Team, Incident Response Team, and IT Recovery Teams.

CO5: Identify vulnerabilities and risks within the IT systems, performing audits and recommending corrective actions to improve recovery strategies.

Textbooks:

1. Susan Snedaker, Chris Rima, "*Business Continuity and Disaster Recovery Planning for IT Professionals",* 2nd Edition, Syngress, 2011.
2. Jamie Watters, "*Disaster Recovery, Crisis Response, and Business Continuity: A Management Desk Reference"*, Apress, 2013.
3. P.W. Singer and Allan Friedman, "*Cybersecurity and Cyberwar: What Everyone Needs to Know",* Oxford University Press, 2014.

Reference Books:

1. Yuri Diogenes, Dr. Erdal Ozkaya, "*Cybersecurity and Cyber Resilience",* Packt, 3rd Edition, 2022.
2. Jason T. Luttgens, Matthew Pepe, and Kevin Mandia, "*Incident Response & Computer Forensics"*, McGraw Hill, 3rd Edition, 2014.
3. Eric C. Thompson, "*Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents"*, Apress, 2018.

Web Links:

1. NIST Cybersecurity Framework: https://www.nist.gov/cyberframework
2. GDPR Compliance Guidelines: https://gdpr.eu/lines
3. CISA Planning: Response & Recovery: https://www.cisa.gov/planning-response-recovery

CYBER INTELLIGENCE			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE672**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**3:0:0**	Exam Hrs	**03**
Total Hrs	**42**	Credits	**03**

Course Learning Objectives:

This course is designed to:

1. Impart knowledge on the building blocks of the threat intelligence life cycle and its associated strategies.
2. Provide methods and approaches for generating intelligence requirements that drive a typical project.
3. Familiarize with incorporating an established threat intelligence framework within the security ecosystem
4. Instill the ability to create a secure and resilient system capable of withstanding potential attacks while ensuring the protection of valuable assets.
5. Familiarize with the types of data required for intelligence gathering and the sources from which it can be collected.

Module 1: Introduction to Cyber Intelligence	**No. of Hrs: 08**

Need for Cyber Intelligence: Intel stories in military, Types of Intelligence: HUMINT, OSINT, SIGINT, COMINT, Intelligence drives operations, Understanding the maneuver warfare mentality.

Textbook 1 - Ch 1

Module 2: Intelligence Development	**No. of Hrs: 08**

The Intelligence Cycle Steps: Planning and direction, Collection, Processing, Analysis and production, Dissemination, Utilization.

Textbook 1 - Ch 2

Module 3: Integrating Cyber Intelligence, Security and Operations	**No. of Hrs: 08**

Developing a strategic cyber intelligence capability, Introduction to Operational Security (OPSEC), Applications of OPSEC in business environments, Cyber Intelligence Program Roles.

Textbook 1 - Ch 3

Module 4: Active Defense and Threat Response	**No. of Hrs: 10**

General principles of Active Defense, Enticement and entrapment in Active Defense, Types of Active Defense, F3EAD Process, F3EAD and the Kill Chain, Applications of F3EAD in the commercial space.

Textbook 1 - Ch 4, 5

Module 5: Threat Intelligence and Collaboration	**No. of Hrs: 08**

Capability Maturity Model, Purpose of Collaboration Capability, Collaboration at the strategic level, Collaboration at the tactical level, Collaboration at the operational level.

Textbook 1 - Ch 6, 7

Course Outcomes:

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

CO1: Apply concepts like maneuver warfare and the OODA Loop to real-world scenarios for analyzing and improving organizational security responses.

CO2: Build and manage intelligence requirements by applying the steps of the intelligence cycle for effective data collection and analysis.

CO3: Use the OPSEC framework to analyze and address potential threats and vulnerabilities within an organization's security operations.

CO4: Utilize the principles of Active Defense and the F3EAD process to design proactive measures against identified threats.

CO5: Make use of threat intelligence data from various sources to build a collaborative and effective threat response framework within security operations.

Textbooks:

1. Wilson Bautista Jr., "*Practical Cyber Intelligence: How action-based intelligence can be an effective response to incidents"*, Packt Publishing, 2018.

Reference Books:

1. Aaron Roberts, "*Cyber Threat Intelligence: The No-Nonsense Guide for CISOs and Security Managers"*, Apress, 2021.
2. Jean Nestor M. Dahj, "*Mastering Cyber Intelligence"*, Packt Publishing, 2022.

Web Links:

1. Threat Intelligence, Why It Matters: https://www.paloaltonetworks.com/cyberpedia/what-is-cyberthreat-intelligence-cti
2. Cyber Threat Intelligence: https://arcx.io/courses/cyber-threat-intelligence-101
3. Mastering Cyber Threat Intelligence: https://www.my-mooc.com/en/book/mastering-cyber-intelligence
4. Mastering Cyber Threat Intelligence, Scratch To Master: https://www.udemy.com/course/mastering-cyber-threat-intelligence-scratch-to-master/?couponCode=LETSLEARNNOWPP

AUGMENTED REALITY AND VIRTUAL REALITY			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE673**	SEE Marks	**50**
Teaching Hrs/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hrs	**50 (26 theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Familiarize the scientific, technical, and engineering aspects of augmented and virtual reality systems.
2. Provide basic AR and VR experiences and use virtual environment technology to solve given problems.
3. Impart knowledge on how VR technology relates to human perception and cognition by evaluating virtual interfaces.
4. Learn the process of how one can interact with a VR world, including the concepts and technologies of VR interaction.

Module 1: Introduction — **No. of Hrs: 06 + 06**

Definition of VR, Modern Experiences, Historical Perspective, Hardware, Software, Human Physiology and Perception.

Textbook 1 - Ch 1, 2

Laboratory Component: (using Unreal Engine)

1. Implement a basic AR video viewing application where users can interact with a virtual video screen and control playback using touch-based interactions.

Module 2: Human-Computer Interaction — **No. of Hrs: 05 + 06**

Human Factors Fundamentals: Introduction, Information Processing, Perception, Cognition, Physical Ergonomics, Guidelines.

General Principles of Human–Computer Interaction: Introduction, Understanding the User Experience, Design Principles and Guidelines, Engineering the User Experience.

Textbook 2 - Ch 3, 4

Laboratory Component:

2. Create an AR application that overlays virtual 3D objects and informational markers in a real-world setting (e.g., a museum, historical site, or campus).

Module 3: Light and Optics & Physiology of Human Vision — **No. of Hrs: 05 + 06**

Light and Optics: Basic Behaviour of Light, Lenses, Optical aberrations, Human Eye, Cameras, Displays The Physiology of Human Vision: Eye Movement, Implications of VR.

Textbook 1 - Ch 4, 5

Laboratory Component:

3. Develop an AR application that overlays animated materials onto virtual objects placed in a real-world environment. The animated materials should visually respond to user interactions and

change based on the object's state (e.g., glowing, pulsing, or color-shifting effects).

Module 4: Visual Perception & Rendering	**No. of Hrs: 05 + 06**

Visual Perception: Perception of Depth, Perception of Motion, Perception of Color.
Visual Rendering: Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates, Immersive Photos and Videos

Textbook 1 - Ch 6, 7

Laboratory Component:

4. Develop an AR application that visualizes a 3D model of the solar system in real-world space.

Module 5: Audio & Evaluating VR Systems and Experiences	**No. of Hrs: 05**

Audio: The Physiology of Human Hearing, Auditory Perception, Auditory Rendering.
Evaluating VR Systems and Experiences: Recommendations for Developers, Comfort and VR Sickness, Experiments on Human Subjects.

Textbook 1 - Ch 11, 12

Course Outcomes:

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

CO1: Outline the working of AR and VR systems under specific scenarios.
CO2: Make use of immersive AR/VR techniques for a given scenario to identify issues in each of them.
CO3: Apply specific design techniques to render the required AR and VR experiences.
CO4: Model solutions for the state-of-the-art AR and VR design problems in industry and academia.

Textbooks:

1. LaValle, M. Steven, "*Virtual Reality",* Cambridge University Press, 2023.
2. LaViola, J.J., JR, Kruijff, E., Bowman, D A., McMahan, R P, & Poupyrev, I, "*3D User Interfaces: Theory and Practice"*, Addison Wesley, 2nd Edition, 2017.

Reference Books:

1. Jerald, Jason, "*The VR book: Human-centered design for Virtual Reality"*, Morgan & Claypool, 2015.
2. George Mather, "*Foundations of Sensation and Perception"*, Psychology Press, 2nd Edition, 2009.
3. Grigore C. Burdea, Philippe Coiffet, "*Virtual Reality Technology"*, Wiley-IEEE Press, 2nd Edition, 2003.
4. Alan B. Craig, William R. Sherman, Jeffrey D. Will, "*Developing Virtual Reality Applications: Foundations of Effective Design",* Morgan Kaufmann Publishers, 2009.

Web Links:

1. Virtual Reality: https://nptel.ac.in/courses/106106138
2. Augmented Reality: https://www.freecodecamp.org/news/augmented-reality-full-course/
3. Intro to AR/VR/MR/XR: Technologies, Applications & Issues: https://www.coursera.org/learn/intro-augmented-virtual-mixed-extended-reality-technologies-applications-issues

REINFORCEMENT LEARNING			
Semester	**IV**	CIE Marks	**50**
Course Code	**23MCPE674**	SEE Marks	**50**
Teaching Hours/Week (L:T:P)	**2:0:2**	Exam Hrs	**03**
Total Hours	**50 (26 Theory + 24 Lab)**	Credits	**03**

Course Learning Objectives:

This course is designed to

1. Provide the key concepts of Reinforcement Learning.
2. Impart knowledge on core Reinforcement Learning algorithms.
3. Instill the ability to learn optimal policies in complex environments.
4. Provide knowledge on Reinforcement Learning techniques that extend beyond standard policy gradient methods.
5. Familiarize with advanced techniques for tackling complex decision-making problems.

Module 1 : Introduction to Reinforcement Learning	**No. of Hrs: 06+06**

Reinforcement Learning (RL), Taxonomy of RL Approaches, Fundamental Concepts in RL. Multi-Arm Bandit Testing, Bandit algorithm, Markov decision processes, Policies and Value Functions, Monte Carlo Policy Generation, Value Iteration with Dynamic Programming.

Textbook 1 - Ch 1, 2

Laboratory Component

1. Implement a simple multi-armed bandit algorithm where an agent interacts with an environment consisting of multiple slot machines.
2. Create a simple grid world environment where an agent moves around to maximize rewards. Implement the agent, actions (move up, down, left, right), states (grid positions), and rewards. Demonstrate a simple reward-based learning process.

Module 2: Temporal-Difference Learning, Q-Learning, and n-Step Algorithms	**No. of Hrs: 05+04**

Formulation of Temporal-Difference Learning, Q-Learning, SARSA, n-Step algorithms, n-Step algorithms on Grid Environments.

Textbook 1 - Ch 3

Laboratory Component

3. Implement the Q-learning algorithm to solve a simple reinforcement learning problem, such as a maze-solving or gridworld navigation task. The agent should update its action-value function Q(s, a) using the Q-learning update rule and learn an optimal policy over time.

Module 3: Deep Q-Networks and Policy Gradient Methods	**No. of Hrs: 05+06**

Deep Q-Learning, Implementing DQN. Policy Gradient Theorem, Policy Functions, Monte Carlo (REINFORCE), REINFORCE with Baseline, Gradient Variance Reduction.

Textbook 1 - Ch 4, 5

Laboratory Component

4. Implement a Deep Q-Network (DQN) where a neural network is used to approximate the action-value function Q(s, a). The agent learns to play an environment using experience replay and a target network to stabilize training.
5. Implement the REINFORCE algorithm, a policy gradient method, where the agent learns a parameterized policy using a neural network and updates it based on the return observed from each episode. The agent learns to maximize the expected reward in an environment.

Module 4 : Beyond Policy Gradients	**No. of Hrs: 05+ 04**

Off-Policy Algorithms, Behavior and Target Policies, Off-Policy Q-Learning, Gradient Temporal-Difference Learning, Greedy-GQ, Off-Policy Actor-Critics, Deterministic Policy Gradients, Deep Deterministic Policy Gradients.

Textbook 1 - Ch 6

Laboratory Component

6. Implement an off-policy Q-learning algorithm to solve a simple grid world environment, where an agent must learn an optimal policy to reach a goal. Use epsilon-greedy exploration to balance exploration and exploitation, and update the Q-values based on an off-policy approach.

Module 5 : Hierarchical and Multi-agent RL	**No. of Hrs: 05+04**

Partially Observable Markov Decision Process (POMDP), Contextual Markov Decision Processes, Hierarchical Reinforcement Learning, Multi-Agent Reinforcement Learning: framework, Single-Agent Algorithms.

Textbook 1 - Ch 8

Laboratory Component

7. Design a hierarchical reinforcement learning program where the agent learns multiple tasks to achieve a larger goal. For example, in a simulated robotic environment, the agent might learn sub-tasks such as "pick up object," "navigate to location," and "place object," which collectively form the larger task of "organize the workspace."

Course Outcomes:

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

CO1: Outline the fundamental concepts of reinforcement learning.
CO2: Identify and implement core Reinforcement Learning algorithms.
CO3: Apply Reinforcement Learning techniques to learn optimal policies in complex and dynamic environments.
CO4: Utilize advanced RL techniques that extend beyond standard policy gradients.
CO5: Make use of advanced RL techniques, to solve complex decision-making problems.

Textbooks:

1. Phil Winder, "*Reinforcement Learning Industrial Applications of Intelligent Agents",* 1st Edition, 2020, O'Reilly Media.

Reference Books:

1. Richard S. Sutton and Andrew G. Barto*, "Reinforcement Learning*: *An Introduction"*, 2nd Edition, November 2018.
2. Laura Graesser, Wah Loon Keng, "*Foundations of Deep Reinforcement Learning:*

Theory and Practice in Python", 2019, Addison-Wesley Professional.
3. Warren B. Powell, "*Reinforcement Learning and Stochastic Optimization"*, 2022, Wiley.

Web Links:

1. Reinforcement Learning Course: https://github.com/upb-lea/reinforcement_learning_course_materials
2. Introduction to Reinforcement Learning: https://www.youtube.com/watch?v=Mut_u40Sqz4
3. Reinforcement Learning: https://www.ibm.com/topics/reinforcement-learning

MAJOR PROJECT PHASE - 2			
Semester	**IV**	CIE Marks	**100**
Course Code	**23MCSE621**	SEE Marks	**100**
Teaching Hours/Week (L:T:P)	**-**	Exam Hrs	**03**
Total Hours	**-**	Credits	**16**

Course Learning Objectives:

This course is designed to

1. Implement the project with the already identified requirements
2. Organize his/her project work in the appropriate manner and present information on the project clearly to all the stakeholders.
3. Ensure that a right solution is obtained as per the requirements of the problem in hand.
4. Enable a student to present his/her project work
5. Present the project work in a technical seminar and document the project work via a project report.

Guidelines

1. A student is expected to carry out the project work in regular consultation with the supervisor/guide to whom he/she was assigned during Major Project Phase-1.
2. Attending Project reviews.
3. Presenting or Publishing the project work in a reputed conference / journal.

Course Outcomes :

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

CO1: Apply disciplined software engineering principles and practices throughout the development of the project work.

CO2: Build the software application as specified in the software requirement specification.

CO3: Choose and acquire the appropriate skills required for a typical project from time to time.

CO4: Use critical thinking and problem solving skills required for developing a software application.

MOOC COURSES			
Semester	**IV**	CIE Marks	-
Course Code	**23MCNM622**	SEE Marks	-
Teaching Hours/Week (L:T:P)	-	Exam Hrs	-
Total Hours	-	Credits	-

Guidelines

- The student should complete one NPTEL Course of 8/12 weeks duration during their second year of MCA program.
- He/She should take up the course in their respective stream of interest which they have already opted for during the second semester.
- The student will be informed before they go for their internships.
- The respective stream coordinators will suggest a list of courses to the students.
- The list of courses will get ratified from the Board of Studies (BOS).
- The student has to select a course that has been suggested by the respective stream coordinator from the authorized list.
- He/She will not be allowed to select any other course apart from the BOS recommended list of courses in their particular stream they have opted.
- In addition, if a student wants to complete more than one NPTEL course, or any other MOOC course it is his/her choice.
- This is apart from one mandatory course of 8/12 weeks duration.
- In case the student is not able to clear the NPTEL course in the first attempt, he/she can take up any other NPTEL course of duration 8/12 weeks in his/her stream of interest that will again be suggested by the respective stream coordinators.
- The course list for this will be provided once the NPTEL releases the list of courses from time to time.