Program Booklet

Department of Computer Science and Engineering

Birla Institute of Technology, Mesra, Ranchi, Jharkhand

Master of Computer Applications

Institute Vision

To become a Globally Recognized Academic Institution in consonance with the social, economic and ecological environment, striving continuously for excellence in education, research and technological service to the National needs.

Institute Mission

To educate students at Undergraduate, Postgraduate, Doctoral, and Post-Doctoral levels to perform challenging engineering and managerial jobs in industry.

• To provide excellent research and development facilities to take up Ph.D. programmes and research projects.

• To develop effective teaching and learning skills and state of art research potential of the faculty.

• To build national capabilities in technology, education and research in emerging areas.

• To provide excellent technological services to satisfy the requirements of the industry and overall academic needs of society.

Department Vision

The department strives to be recognized globally for outstanding education and research, leading to excellent professionals and innovators in the field of Computer Science and Engineering, who can positively contribute to the society.

Department Mission

- To impart quality education and equip the students with strong foundation that could make them capable of handling challenges of the new century.

- To maintain state of the art research facilities and facilitate interaction with world's leading universities, industries and research organization for constant improvement in the quality of education and research.

Programme Educational Objectives (PEOs): Master of Computer Applications

PEO 1: To build scalable, high-quality applications using modern engineering practices.
PEO 2: Develop ethical leadership and professional integrity while fostering a collaborative team spirit.
PEO 3: To be prepared for advanced studies, teaching, and research while fostering innovation to contribute to national growth.

PROGRAM OUTCOMES (POs) for MCA (MASTER OF COMPUTER APPLICATIONS)

PO1 (Foundation Knowledge):
Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO2 (Problem Analysis):
Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

PO3 (Development of Solutions):
Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO4 (Modern Tool Usage):
Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO5 (Individual and Teamwork):
Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO6 (Project Management and Finance):
Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO7 (Ethics):
Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.

PO8 (Life-long learning):
Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

PROGRAMME COURSE STRUCTURE (ALL SEMESTERS)

BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI, JHARKHAND COURSE STRUCTURE – *Proposed from Monsoon 2026* Based on New Education Policy (NEP 2020) Model Recommended scheme of study for MCA Programme									
Sl. No.	**Semester of Study (Recomm-ended)**	**Category of Course**	**Course Code**	**Course Name**	**Mode of Delivery and Credits L-Lecture, T-Tutorial, P-Practical, SL-Self-Learning**				**Total Credits**
					L (Periods /Week)	T (Periods /Week)	P (Periods /Week)	SL (Periods /Week)	
		THEORY							
1.1	Semester I	Core1 (Th)	CA26501	Understanding OOP using Java	3	0	0	0	3
1.2		Core2 (Th)	CA26503	Data Structures	3	0	0	0	3
1.3		Core3 (Th)	CA26505	Concepts of Database Management System	3	0	0	0	3
1.4		Core4 (Th)	CA26507	Formal Language and Automata Theory	3	0	0	0	3
1.5		Core5 (Th)	CA26509	Mathematics for Computing	3	0	0	0	3
		LABORATORY							
1.6		Core1 (P)	CA26504	Data Structures using Java Lab	0	0	3	0	1.5
1.7		Core2 (P)	CA26506	Database Management System Lab	0	0	3	0	1.5
1.8		Skill1	CA26510	Web Application Development Skills using HTML and CSS	0	0	2	2	2
TOTAL (THEORY + LAB + SKILL)									**20**
		THEORY							

2.1	Semester II	Core1 (Th)	CA26511	Operating Systems	3	0	0	0	3
2.2		Core2 (Th)	CA26513	Analysis of Algorithms	3	0	0	0	3
2.3		Core3 (Th)	CA26515	Python Programming	2	0	0	0	2
2.4		Core4 (Th)	CA26517	Software Design Patterns	3	0	0	0	3
2.5		Core5 (Th)	CA26519	Data Communication and Computer Networks	3	0	0	0	3
2.6		Elec1 (Th)	CA265XX	PE 1	3	0	0	0	3
		LABORATORY							
2.7		Core1 (P)	CA26516	Python in Linux	0	0	3	0	1.5
2.8		Core2 (P)	CA26520	Data Communication and Computer Networks Lab	0	0	3	0	1.5
2.9		Skill1	CA26522	Learning Power BI Applications	0	0	2	2	2
TOTAL (THEORY + LAB + SKILL)									22
GRAND TOTAL FOR FIRST YEAR									42
Exit After 1st Year with Post Graduate Diploma in Computer Applications									
2.10	**PG Diploma in Computer Applications**		**Research/ Application Project**				**4**	**Total Credit 46**	
3.1	Semester III	Core1 (Th)	CA26601	Machine Learning	3	0	0	0	3
3.2		Core2 (Th)	CA26603	Cryptography	3	0	0	0	3
3.3		Elec2 (Th)	CA266XX	PE 2	3	0	0	0	3
3.4		Elec3 (Th)	CA266XX	PE 3	3	0	0	0	3
3.5		OE	XX26XXX	Open Elective/ MOOC (Technology Entrepreneurship or relevant courses)	3	0	0	0	3
3.6		**LABORATORY**							
3.7		Core1 (P)	CA26602	Machine Learning Lab	0	0	3	0	1.5
3.8		Elec1 (P)	CA266XX	PE 3 Lab	0	0	3	0	1.5

3.9		Project	CA26600	Research/ Application Project					6
TOTAL (THEORY + LAB + PROJECT)									24
4.1	Semester IV	Core1	CA26660	Research Project/ Industry Internship					20
TOTAL (PROJECT + VIVA)									20
GRAND TOTAL FOR SECOND YEAR									44
GRAND TOTAL FOR MCA									86
Lateral Entry After First Year (Students must cover at least 70% of the Syllabus of regular First year courses) (Academic Eligibility criteria for lateral entry are BCA 4 years, B.Sc. Honours in CS or IT (4 Years) and PGDCA)									

List of Program Electives

PE	Track	Course Code	Course Name
1	**Track: Core Courses**	CA26521	Computer Organization & Architecture
		CA26523	Multimedia Technologies
		CA26525	Data Compression
		CA26527	Digital Signal Processing
		CA26529	Graph Theory
		CA26531	Modelling and Solving Optimization Problems
		CA26533	Fuzzy Computing
	Track: Modern Computing Paradigm	CA26535	Data Warehouse and Data Mining
		CA26537	Information Retrieval
		CA26539	Data Visualization using Python
	Track: Web Application	CA26541	Introduction to NoSQL Backends
		CA26543	Frontend Design 1
		CA26545	Introduction to Web Scraping
2	**Track: Core Courses**	CA26605	Image Processing
		CA26607	Cloud Computing
		CA26609	Network Analysis
	Track: Modern Computing Paradigm	CA26611	Evolutionary Computing
		CA26613	Reinforcement Learning
		CA26615	Natural Language Processing
	Track: Web Application	CA26617	Modern Web Architecture
		CA26619	Web Security
		CA26621	Frontend Design 2
	Track: Advance	CA26623	Advanced Database Systems
		CA26625	Bioinformatics

	d Modern Computing	CA26627	Retrieval Augmented Generation
3	**Track: Core Courses**	CA26629	Internet of Things
		CA26630	Embedded System Lab
		CA26631	Shell Scripting
		CA26632	Shell Scripting Lab
		CA26633	Mobile Application Development using Flutter
		CA26634	Mobile Application Development using Flutter Lab
		CA26635	Software Testing and Version Control
		CA26636	Software Testing and Version Control Lab
		CA26637	Advanced Programming in Java
		CA26638	Advanced Programming in Java Lab
	Track: Modern Computing Paradigm	CA26639	Deep Learning
		CA26640	Deep Learning Lab
		CA26641	Large Language Models
		CA26642	Large Language Models Lab
		CA26643	Generative AI
		CA26644	Generative AI Lab
		CA26645	Computer Vision
		CA26646	Computer Vision Lab
	Track: Web Application	CA26647	Introduction to Django
		CA26648	Full Stack using Django Lab
		CA26649	Introduction to Laravel
		CA26650	Full Stack using Laravel lab
		CA26651	Frontend Engineering using React JS
		CA26652	Full Stack using React JS Lab
	Track: Advanced Modern Computing	CA26653	Agentic AI Systems
		CA26654	Agentic AI Systems Lab
		CA26655	Prompt Engineering
		CA26656	Prompt Engineering Lab
		CA26657	Quantum Computing
		CA26658	Quantum Computing Lab

MCA COURSE STRUCTURE ALIGNING WITH BLOOM'S TAXONOMY

Progression of Cognitive Skills from Knowledge Acquisition to Innovation & Impact

BLOOM'S TAXONOMY (COGNITIVE DOMAIN)

L1 Remember	L2 Understand	L3 Apply	L4 Analyze	L5 Evaluate	L6 Create
Recall facts and basic concepts	Explain ideas and concepts	Use knowledge in new situations	Examine and break information	Make judgments based on criteria	Produce new ideas or solutions

SEMESTER (LEVEL FOCUS)	COURSES (Theory / Practical / Skills / Project)	ASSESSMENT ALIGNMENT (Examples)
SEMESTER I (L1–L3 Focus) Foundation Building	• Understanding OOP using Java • Data Structures • DBMS Concepts • Formal Language & Automata Theory • Mathematics for Computing • Concepts of DBMS • Mathematics for Computing • Web Application Development Skills (HTML & CSS) • Data Structures using Java Lab • DBMS Lab • Web Development (HTML & CSS)	• Quizzes (L1–L2) • Assignments (L2–L3) • Mid Sem Exam (L2–L3) • Lab Performance (L3) • End Sem Exam (L2–L3)
SEMESTER II (L2–L4 Focus) Application & Analysis	• Operating Systems • Analysis of Algorithms • Python Programming • Software Design Patterns • Data Communication & Networks • Python in Linux Lab • DCN Lab • Power BI Applications (Skill Course) • Analysis of Algorithms • Design Patterns – Case Analysis • Mini Projects • Elective 1	• Quizzes (L2) • Assignments (L3) • Mid Sem Exam (L3–L4) • Case Study / Tutorial (L4) • Lab & Skill Evaluation (L3–L4) • End Sem Exam (L3–L4)
SEMESTER III (L3–L6 Focus) Advanced Learning & Problem Solving	• Machine Learning • Cryptography • Elective 2 • Elective 3 • Open Elective / MOOC • ML Lab • Elective Lab • Research/Application Project (Part I) • Case Studies & Paper Analysis • Project Evaluation • Technical Seminar • Model Performance Evaluation • Research Report • Research/Application Project – Innovation • Prototype Development • Solution Design • Project Presentation	• Assignments (L3–L4) • Case Study / Paper Review (L4) • Mid Sem Exam (L3–L4) • Project Work (L4–L6) • Seminar (L5) • End Sem Exam (L4–L5)
SEMESTER IV (L5–L6 Focus) Research, Innovation & Professional Excellence	• Industry Internship / Research Project Evaluation • Progress Presentation • Dissertation / Thesis • Final Project Development • Viva Voce • Industrial Evaluation • Innovation & Deployment	• Project Report (L5–L6) • Viva Voce (L6) • Industrial Evaluation (L5–L6) • Innovation & Impact (L6)

PROGRESSION OF LEARNING ACROSS THE MCA PROGRAM

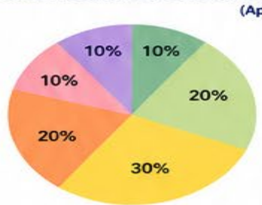
Remember (L1) → Understand (L2) → Apply (L3) → Analyze (L4) → Evaluate (L5) → Create (L6)

(From Knowledge Acquisition to Innovation & Impact)

COURSE CATEGORIES AND BLOOM LEVEL EMPHASIS

Core Theory Courses	L1 – L4
Laboratory Courses	L3 – L6
Skill Courses	L3 – L5
Elective Courses	L3 – L6
Research Project	L4 – L6
Internship / Industry Project	L5 – L6

OVERALL PROGRAM BLOOM DISTRIBUTION (Approx.)



L1 Remember	10%
L2 Understand	20%
L3 Apply	30%
L4 Analyze	20%
L5 Evaluate	10%
L6 Create	10%

DELIVERY METHODS (Aligned to Bloom Levels)

- L1 – Lectures, Reading Materials
- L2 – Interactive Lectures, Discussions
- L3 – Tutorials, Labs, Assignments
- L4 – Case Studies, Problem Solving, Code Analysis
- L5 – Seminars, Paper Reviews, Project Evaluation
- L6 – Projects, Hackathons, Research, Innovation Labs

This framework ensures a systematic progression of cognitive skills across the MCA program in alignment with NEP 2020, Outcome Based Education (OBE) and industry expectations.

COURSE INFORMATION SHEET

Course Code: CA26501

Course Title: Understanding OOP using Java

Pre-requisite(s): Basic Programming

Co-requisite(s): Data Structures using Java Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Shamama Anwar, Dr. Jamimamul Bakas

COURSE OBJECTIVES

This course enables the students to:

1. To introduce core Java programming constructs and the foundational principles of Object-Oriented Programming (OOP)
2. To understand object lifecycles, memory management tools, and secure class design for solving complex problems.
3. To demonstrate how to build reusable, modular software using inheritance hierarchies, interfaces, and packages
4. To explain the mechanisms of exception handling, file processing, and basic text parsing for reliable applications.
5. To enable the students to use advanced regular expressions and multithreading techniques for safe, concurrent task execution.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Learn various Java constructs and being able to develop Java applications using object-oriented programming principles.
CO2	Build and design Java programs using object lifecycles, memory tools, and safe class structures to solve complex coding problems
CO3	Build reusable and modular Java applications by using inheritance hierarchies, interfaces, and packages to manage data
CO4	Build reliable Java applications with exception handling, file processing and regular expression.
CO5	Build efficient Java programs that use regular expressions for text pattern matching and multithreading tools for safe, concurrent task execution.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Object-Oriented Programming Concepts Benefits of OOP, Applications of OOP, Properties of OPP: objects, classes, methods and message communication, abstraction, encapsulation, inheritance, polymorphism. Structure of JAVA Program, Simple Java Program, JVM Architecture and memory management. Data types: primitive and reference, Wrapper class, Array: 1D, 2D, multi-dimensional, jagged arrays, Basic concepts of classes and Objects, String class	8
Module 2: Classes, Objects and Methods Method Overloading, Constructor, Constructor Overloading, Recursion, Pass by value v/s pass by reference, final, this and static keyword, finalize() method, Access modifiers, Nested class, Inner class, Abstract class	8
Module 3: Inheritance, Interface, Package and Java Collections Use of Inheritance, different types of Inheritance, Object class, constructor in inheritance, method overriding, super keyword, final class, Dynamic method dispatch, Interface, Package, Introduction to Java Collections Framework.	8
Module 4: String, Exception and File handling String, StringBuffer, Fundamentals of Exception Handling, Types of Exceptions: Checked and Unchecked, Try-Catch, Throw, Throws, Finally blocks, Designing Custom Exceptions, File Handling: FileReader, FileWriter, BufferedReader, BufferedWriter, Serialization & Deserialization	8
Module 5: Regular Expression and Multithreaded Programming Regular expression: regex, Pattern, Macther class. Multithread programming: Introduction to Threads and Multithreading, Thread Lifecycle and States, Creating Threads: Extending Thread Class, Implementing Runnable Interface, Synchronization, Thread communication, Deadlock	8

Text Books:

1. Herbert Schildt, Dr. Coward, Danny, Java: The Complete Reference, 13th Edition, McGrawHill Publications, 2024.

Reference Books:

1. Cay S. Horstmann, Programming with Java, Beginner to Advanced, 7th Edition, Wiley, 2024

2. Deitel P., Deitel H., Java How to Program, Pearson Publications, 2016.

3. Wu C. T, Object Oriented Programming in Java, McGrawHill Publications, 2009
4. Seth, Anita, and B. L. Juneja, Java: One Step Ahead, Oxford University Press, New Delhi, 2017

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes) (15 marks each)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Board / Slides
CD2	Tutorial / Assignments
CD3	Self-Learning through NPTEL / online materials
CD4	Coding demonstration in class
CD5	Mini projects
CD6	Interactive Lectures, Flipped Classroom Activities
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	1	0	1	0	1
CO2	3	3	3	2	0	1	1	1
CO3	3	3	3	3	0	2	1	1
CO4	3	3	3	3	0	2	1	1
CO5	3	3	3	3	0	2	1	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD4
CO4	CD1, CD2, CD3, CD4, CD5
CO5	CD1, CD2, CD3, CD4, CD5

COURSE INFORMATION SHEET

Course Code: CA26503

Course Title: Data Structures

Pre-requisite(s): Basic Programming

Co-requisite(s): Data Structures using Java Lab

Credits: 03 L: 03 T:0 P:0

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Kumar Rajnish, Dr. Sandip Ghosal

COURSE OBJECTIVES

This course enables the students to:

1. Explain how to use and implement data structures in real-world situations.
2. Use your knowledge of various data structures to build a program.
3. Develop the logical capability to efficiently store and retrieve data.
4. Gain an understanding of how graph theory may be used to solve practical issues.
5. Create and build object-oriented software while understanding data abstraction.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand and determine the complexity of problems for efficient problem solving.
CO2	Apply linked list operations to implement linear data structures.
CO3	Implement stack and queue data structures to solve computational problems.
CO4	Apply and analyze tree and graph algorithms for efficient data organization and traversal.
CO5	Evaluate searching and sorting algorithms based on their performance and suitability.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Algorithms and Introduction and Array Definition, Pseudo-code Representation, Time complexity and space complexity, efficiency of algorithms - Big O-notation - Omega notation and Theta. Recurrence relations, solving recurrences using recursion trees and back substitution method Definition of data structure, data structure operations. Linear arrays, Representation of linear arrays in memory, Address calculation of using row and column major ordering, Traversing linear arrays, Inserting and Deleting, Multidimensional arrays: Representation of Two-Dimensional arrays in memory.	8
Module 2: Linked Lists Introduction, single linked list, representation of a linked list in memory, Operations on a single linked list, reversing a single linked list, Circular linked list and doubly linked list.	7
Module 3: Stacks and Queues Introduction, Array representation of stacks, Linked representation of stacks, Polish notation, Evaluation of a Postfix Expression, Transforming Infix Expressions into Postfix Expressions. Definition, Array representation of Queues, Linked representation of Queues, Circular queues, Priority queues and D-Queue. Implement stack using linked list, Implement Queue using Linked List. Other applications of stacks and queues.	9
Module 4: Trees and Graphs Tree Terminology, Binary Tree, Tree Traversals, Binary Search Tree, Height Balanced Trees: AVL tree, B-tree and B+ Trees. Representation of Graphs, Searching in Graphs – breadth first search and its applications, depth first search and its applications, Shortest paths, Spanning Trees.	8
Module 5: Searching and Sorting Linear Search, Binary Search, Transpose Sequential Search, Interpolation Search. Merge sort, Quick sort, Heap: Priority Queues, Array Implementation of Heaps, Construction of Heaps, Heap Sort. Comparing Sorting Algorithms. External Sorting methods.	8

Text Books:

1. Goodrich, M., Tamassia, R. and Mount D, Data Structures and Algorithms in C++/Java, 2nd Edition, Wiley, 2016.

2. Reema Thareja, Data Structure and algorithms, 1st Edition, Oxford University Press, 2025, ISBN: 9789367252673.

3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, 3rd Edition, Prentice-Hall of India Learning Pvt. Ltd. 2010.

Reference Books:

1. Seymour Lipscutz, Data Structures, 2ndEdition, Schaum's Outlines, Tata McGraw Hill, 2017.

2. Ellis Horowitz, Sartaj Sahani and Susan Anderson Freed, Fundamentals of Data Structures, 2nd Edition, Universities Press.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors/Smart Board
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and Internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	3	1	1	0	3
CO2	3	2	2	3	1	1	0	3
CO3	3	2	2	3	1	1	0	3
CO4	3	2	2	3	1	1	0	3
CO5	3	2	2	3	1	2	0	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26505

Course Title: Concepts of Database Management System

Pre-requisite(s): None

Co-requisite(s): Database Management System Lab

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Shashank Pushkar, Dr. Saikat Chakraborty

COURSE OBJECTIVES

This course enables the students to:

1. Observe how the real-world data is stored, retrieved, and communicate under the DBMS environment
2. Design a logical model which has Relations.
3. Apply the query for the modification of the system.
4. Develop a conceptual design which allows to avoid anomalies in superior's data.
5. Discuss a system which allows to restrict the uncontrolled exaction and provide rigorous variation of the task.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Describe various data models and schemas used in database management systems.
CO2	Explain the fundamental concepts, data definitions and query processing tasks in relational query languages.
CO3	Recognize database design theory and evaluate functional dependencies and normal forms in databases.
CO4	Formulate the operations of transaction and concurrent query processing tasks to obtain the correct results even under strict time constraints.
CO5	Interpret the foundational concepts of distributed databases. Illustrate several techniques related to transaction management and query processing in distributed database management systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction and Conceptual Modelling Characteristics of the Database Approaches and Database Users, Data Models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database languages, Database Architectures, High-Level Conceptual Data Models for Database Design, Entities and Entity Sets, Relationships and Relationship Sets, Keys, Mapping, Cardinalities, Diagram, Generalization and Specialization, Aggregation.	8
Module 2: Relational Model: Concepts, Constraints, Languages, Design and Programming Relational Model Concepts, Relational database Schemas, Update operations, Transactions, Dealing with constraint violations, Operations of Relational Algebra, Additional Relational Operations. ER to Relational Mapping, Basic queries in SQL, More complex SQL Queries, Views, Authorization, Assertion, Triggers.	8
Module 3: Database Design Theory and Methodology Design Guidelines for Relational Schema, Functional Dependencies, Decomposition Using Functional Dependencies. Normalization using functional Dependencies, General Definition of First, Second, Third and Forth Normal Form. Boyce-Codd Normal Form (BCNF), Multivalued and join dependencies, 4^{th} Normal Form, DKNF.	8
Module 4: Transaction Processing Concepts and Concurrency Control Techniques Transaction Processing, Desirable Properties of Transactions, Transaction State, Characterizing Schedules based on Recoverability and Serializability. Lock-Based Protocols, Timestamp-Based Protocols, Deadlock Handling, Recovery Concepts.	8
Module 5: Distributed Databases and Client-Server Architectures Concepts and Types of Distributed databases, data fragmentation, Replication and Allocation Techniques for Distributed Database Design, Query Processing in Distributed Databases, Overview of Concurrency Control and Recovery in Distributed Databases, An Overview of 3-Tier Client-Server Architecture.	8

Text Books:

1. Elmasri Ramez, & Navathe S.B., Fundamentals of Database Systems, 5th Edition, Pearson Education, 2013.

Reference Books:

1. Silberschatz A., &Korth H., Database Systems Concepts, 5th Edition, McGraw Hill Higher Education, 2005.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
Semester End Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment (s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	0	3	2	1	3	1	1
CO2	2	0	2	2	2	2	1	1
CO3	2	1	2	3	3	3	2	1
CO4	3	0	2	2	2	2	1	1
CO5	1	1	1	2	3	2	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26507

Course Title: Formal Language and Automata Theory

Pre-requisite(s): Basic Mathematics and Methods of Proofing

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Anup Kumar Keshri, Dr. Supreeti Kamilya

COURSE OBJECTIVES

This course enables the students to:

1. Design and represent a system in its abstract form with its mathematical modelling.
2. Represent infinite set of strings in finite way to use them in finite machines.
3. Design mathematical model of languages.
4. Design of abstract machines of different levels of computational capabilities.
5. Learn the classes of problems and limitation of computing.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Produce mathematical models of computing devices
CO2	Classify different types of languages and process them with suitable machine
CO3	Construct and add new features to programming languages
CO4	Design abstract machines which are used in design of Compilers
CO5	Analyze real life computational problems and classify them into P, NP, NP-C and NP-Hard problems

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Automata Alphabets, Strings, Languages, Abstract and Mathematical model of Finite Automata, Deterministic and Nondeterministic Finite Automata (DFA & NFA), ε-NFA, conversion from ε-NFA to NFA to DFA, Minimization of states of DFA, Finite automata with output, Application of Finite Automata	10
Module 2: Regular Expressions (RE) Regular expressions definition, representation and basic operations, Algebraic laws on Regular Expression, Constructing NFA from RE, Pumping Lemma for Regular Language, Closure properties of Regular Languages, Applications of Regular Expressions	6
Module 3: Concepts of Grammar Mathematical Definition of Grammar, Chomsky classification of Grammar, Designing Regular grammar from DFA, Context Free Grammar (CFG), Operation on CFG: Identification and removal of useless symbols and unit productions, Closure properties of Context Free Languages, Chomsky and Greibach Normal Form of CFG's, Applications of CFGs.	8
Module 4: Pushdown Automata (PDA) and Parsing Mathematical definition and design, Types of Pushdown Automata (DPDA and NPDA), Converting CFG to PDA, PDA and Parsing: Derivation, Parse Tree, Ambiguous and Unambiguous Grammar, Simulation of derivation using PDA (Parsing)	8
Module 5: Turing Machine (TM) and Types of Problems Basics and Mathematical Definition of TM, Example problems, Variants of TM: Multi Tape, Multi head TM's and their equivalency, Halting Problem of TM, Universal TM, Enumerable Languages, Solvable and Unsolvable problems, Post Correspondence Problems (PCP), Classes of Problems: P, NP, NP-C and NP-Hard.	8

Text Books:

1. Martin John, Introduction to Languages and the Theory of Computation, 4th Edition, Mc GrawHill, 2010
2. Hopcroft John E. And Ullman Jeffrey D., Introduction to Automata Theory, Languages & Computation, 3rd Edition, Narosa, 2008.

Reference Books:

1. Mishra K.L.P & Chandrasekharan N., Theory of Computer Science, Prentice Hall India Learning Private Limited, 2006.
2. Lewis H. R. and Papadimitrou C. H, Elements of the theory of Computation, Prentice Hall India, 1997.
3. Greenlaw R. and Hoover H. J., Fundamentals of the Theory of Computation Principles and Practice, Morgan Kaufmann Publishers, 1998
4. Kulkarni V., Theory of Computation, Oxford University Press, 2014
5. Azad S. K., Theory of Computation: An Introduction to Automata, Formal Languages and Computability, Dhanpat Rai & Co.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Teaching aids/Seminars

Code	Course Delivery Method
CD4	Mini Projects
CD5	Expert talks
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	3	3	3	0	2	0	0
CO2	2	3	3	3	0	2	0	0
CO3	3	3	3	3	0	2	0	0
CO4	2	2	2	2	0	2	0	0
CO5	3	3	3	3	0	1	0	0

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5, CD6, CD7
CO2	CD1, CD4, CD6, CD7
CO3	CD1, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6
CO5	CD1, CD2, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26509

Course Title: Mathematics for Computing

Pre-requisite(s): None

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. B. K. Sarkar, Dr. S. Vijayvargiya

COURSE OBJECTIVES

This course enables the students to:

1. Understand the basic concepts and techniques computing.
2. Use the concept of mathematics to structure problems and solve those.
3. Develop mathematical thinking and problem-solving skill.
4. Provide the foundations of probabilistic and statistical analysis.
5. Apply the concept of mathematical functions in Machine Learning.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Demonstrate skills in solving mathematical problems.
CO2	Analyze problems and identify the computing requirements appropriate to its solution.
CO3	Understand basic concepts in probability theory and statistical analysis
CO4	Attain sufficient knowledge in the context of Machine learning, Data Science, Computational problems.
CO5	Apply mathematical concepts to model, solve and analyse real-world situations.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module1: Mathematical Foundations for Computing Introduction to Set, Relation: order pair and cartesian product, Function: Types of function, Some Special Functions: Growth, Hash, Floor, Ceil, Step, Sigmoid, Tanh, ReLU, Softmax, Convex and non-convex; Logic and *operators*.	8
Module 2: Abstract and Linear Algebra Introduction to Group, Ring, Field, Vector: Basic operations on Vectors and Vector norms. Types of Matrices and basic operations, Concept on Determinants, Solving set of Linear equations - Direct and Iterative: Gauss Elimination and Jacobi methods.	8
Module 3: Combinatorics and Recurrence Relation Combinatorics: Enumerative, Existential and Constructive, Basic principles on counting: Addition, Multiplication, Inclusion-Exclusion and Pigeonhole, Factorial notation for Permutations and Combinations, Counting of arrangements using Recurrence Relation (RR), Methods to solve RR: Iteration, Characteristics function, Generating function, and Master Theorem.	8
Module 4: Probability Theory and Random Variables Basic Concepts of Probability, Conditional probability, Total probability, Bayes' theorem, Random variable (RV): Discrete and Continuous, Probability Distribution Function: Probability mass function (PMF) and Probability Density Function (PDF), Expected value of RV, Concept on Entropy measure, Probability distribution: Discrete and Continuous, Binomial, Poisson and Normal distribution.	8
Module 5: Statistical Measures and Correlation Analysis Basic Statistics: Measure of central tendency: Moments, Expectation, Dispersion, skewness, Correlation measure: Covariance and Correlation for multivariate: Pearson Correlation Coefficient (PCC) and Spearman's rank correlation coefficient (SRCC), Linear Regression.	8

Text Books:

1. Kenneth H. Rossen, Discrete Mathematics and Its Applications, Mc Graw Hill Education, 8th Edition, 2019.
2. H. C. Saxena, J. N. Kapoor, Mathematical Statistics, S Chand Publishing, 2010.

Reference Books:

1. Eric Lehman, F Tom Leighton, Albert R Meyer, Mathematics for Computer Science, Samurai Media Limited, 2017

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Teaching aids/Seminars
CD4	Mini Projects
CD5	Expert talks
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	0	0	0	3
CO2	3	3	3	1	0	0	0	3
CO3	3	2	2	1	0	0	0	3
CO4	3	3	3	3	0	1	2	3
CO5	3	3	3	3	2	1	2	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6, CD7
CO2	CD1, CD2, CD3, CD6, CD7
CO3	CD1, CD2, CD3, CD6
CO4	CD1, CD3, CD5, CD6, CD7
CO5	CD1, CD2, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26504

Course Title: Data Structures Using Java Lab

Pre-requisite(s): Programming Concepts

Co-requisite(s): Data Structures, Understanding OOPs using Java

Credits: 1.5 L: 0 T:0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Kumar Rajnish, Dr. Shamama Anwar

COURSE OBJECTIVES

This course enables the students to:

1. To evaluate the effects of algorithm design techniques and data structure selection on program performance.
2. To select the best algorithm design technique and data format for a given application.
3.To solve issues utilizing data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, binary search trees, and graphs, and write programs for these solutions.
4.Analyze and compare different algorithms.
5.Create and build object-oriented programs while understanding data abstraction.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Select an appropriate data structure for a computation based problem.
CO2	Design and evaluate the time and space efficiency of different data structures.
CO3	Analyze the runtime execution of previously learned sorting algorithms, including selection, merge sort, heap sort, and quick sort.
CO4	Have practical understanding about the applications of data structures.
CO5	Justify the selection of data structure for a specific situation.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Object-Oriented Java and Arrays Programs based on review of OOPs and Memory Management in Java, Linear Arrays and Complexity Analysis.	12
Module 2: Linked Lists Programs based on Linked Lists i.e. Dynamic Memory Management, various applications	9
Module 3: Stacks and Queues Program based on Stacks and Queues and their applications	6
Module 4: Trees Program based on Tress i.e. Creation of BST, Traversal, Operations and some application of BST	6
Module 5: Graphs Program based on graph i.e. their representation, traversal and finding shortest path	6

Text Books:

1. Goodrich, M., Tamassia, R. and Mount D, Data Structures and Algorithms in C++/Java, 2nd Edition, Wiley, 2016.
2. Robert Lafore., Data Structures & Algorithms in Java, AMS Publisher, ISBN: 1571690956, S, 2002.

Reference Books:

3. Mark Allen weiss, Data Structures & Problem Solving Using Java, Fourth Addition, Pearson, Addison Wisley. 1997, ISBN-13: 978-0-321-54140-6.
4. Data Structures and Algorithms in java, 3rd edition, A. Drozdek, Cengage Learning, 2008.
5. Data Structures with Java, J. R. Hubbard, 2nd edition, Schaum's Outlines, Mc Graw Hill, 2009.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Quiz	10
Day to Day Evaluation	30
Viva	20
Total	60

End Semester Assessment	% Distribution
Quiz	10
Performance (End Semester Examination)	30
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aid
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	2	0	0	1	3
CO2	3	3	3	2	0	0	1	3
CO3	3	3	3	2	0	0	1	3
CO4	3	3	3	2	0	0	1	3
CO5	3	3	3	2	0	0	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD2, CD3
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD7
CO5	CD1, CD2, CD3

COURSE INFORMATION SHEET

Course Code: CA26506

Course Title: Database Management System Lab

Pre-requisite(s): None

Co-requisite(s): Database Management System

Credits: 1.5 L:0 T:0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Saikat Chakraborty, Dr. S. Pushkar

COURSE OBJECTIVES

This course enables the students to:

1. Learn the concepts of database models using Entity-Relationship diagrams and practical implementation of designing databases.
2. Understand and implement DDL and DML commands.
3. Implement PL/SQL, CURSORS, Exceptions, Composite Data types.
4. Execute stored procedures, functions and database Triggers.
5. Design Embedded queries and develop database applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Design ER diagram of a production level database and conversion of the schemas into tables.
CO2	Demonstrate real life databases with appropriate relations, constraints, and domain values.
CO3	Demonstrate efficient retrieval of data from single or multiple tables and views.
CO4	Understand and execute advanced queries and procedures such as trigger, PL/SQL.
CO5	To design and implement real life database applications and use as embedded SQL.

SYLLABUS

Assignments	NO. OF LECTURE HOURS
Module 1: Design and implementation of ER Diagram Use database administration software or their FOSS alternatives to create ER diagrams.	6
Module 2: Implementation of DDL and DML statements Use SQL for creating databases, Tables, Fields, Relationships, key and domain constraints, insertion of data, alter table commands etc.	6
Module 3: Implementation of Extended Query Language Use SQL for simple and advanced querying of databases including where clause, field and aggregate functions, nested queries, group by operations, joins, materialized and non-materialized view creation etc.	9
Module 4: Designing Triggers, PL/SQL Blocks Use SQL to create indexes, triggers (on update, insert, delete), cursors, PL/SQL blocks.	9
Module 5: Implementation of Admin Tasks and Embedded Queries Perform other database administrative tasks like user management including grant, revoke, transaction management (COMMIT, SAVEPOINT, ROLLBACK, BEGIN and END transaction), application of embedded queries.	9

Text Books:

1. Elmasri Ramez, Navathe S.B., Fundamentals of Database Systems, 7th Edition, Pearson Education, 2017.

Reference Books:

1. A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 7th Edn, Tata Mc-Graw Hill, New Delhi, 2020.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance & Lab files	30
Quiz	10
Viva	20
Total	60

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aid
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	2	0	3	2	3
CO2	3	3	3	3	0	3	2	3
CO3	3	2	3	3	0	3	3	3
CO4	3	3	3	2	0	3	3	3
CO5	3	3	3	3	2	3	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD6
CO2	CD1, CD2, CD3, CD6
CO3	CD1, CD2, CD3, CD6
CO4	CD1, CD2, CD3, CD6
CO5	CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26510

Course Title: Web Application Development Skills using HTML and CSS

Pre-requisite(s): None

Co-requisite(s): None

Credits: 2 L: 0 T: 0 P: 2 SL: 2

Class Schedule per Week: Class schedule per week: 04

Class: MCA

Semester / Level: I / V

Branch: MCA

Name of the Teacher: Dr. Anurag Joshi, Prof. Abhijit Mustafi

COURSE OBJECTIVES

This course enables the students to:

1. Explain the fundamentals of HTML5, semantic markup, hyperlinks, and webpage structure. (BL2)
2. Develop web pages incorporating multimedia elements, tables, and hyperlinks using HTML5. (BL3)
3. Design interactive web forms with appropriate validation and input controls. (BL3)
4. Apply CSS selectors, cascading rules, inheritance, and styling techniques to develop attractive web pages. (BL3)
5. Design complete static websites using HTML5 and CSS with appropriate layouts, typography, colors, and responsive design principles. (BL6)

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamentals of HTML5, semantic markup, hyperlinks, and webpage structure. (BL2)
CO2	Develop web pages incorporating multimedia elements, tables, and hyperlinks using HTML5. (BL3)
CO3	Design interactive web forms with appropriate validation and input controls. (BL3)
CO4	Apply CSS selectors, cascading rules, inheritance, and styling techniques to develop attractive web pages. (BL3)
CO5	Design complete static websites using HTML5 and CSS with appropriate layouts, typography, colors, and responsive design principles. (BL6)

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to HTML Starting off with HTML, How HTML Tags Work, Adding Comments, Structuring an HTML Page, the <meta> Tag, What Is Semantic Markup? HTML Text Formatting, Paragraphs and Headings, Lists, Quoting a Block of Text, Formatting Text Inline, Marking Up Code, Link Markup, URL Structure, Internal vs. External Linking, Relative vs. Absolute Linking Other Types of Links, Link Targets.	4
Module 2: Hyperlinks, Images, and Multimedia Link Markup, URL Structure, Internal vs. External Linking, Relative vs. Absolute Linking, Other Types of Links, Link Targets, How Does Media Work on the Web? Images, Adding Images to a Page, Responsive Images: Considering Different Devices and Connections, the <picture> Element, Embedding Video, Embedding Audio, Storing Multimedia Files.	9
Module 3: HTML Tables and Web Forms Tables, Description Lists, Interacting with Webpages, how a Web Form Works, Components of an HTML Form, The <form> Element, Form Fields, Labelling Fields, Setting Up a Basic Form, Creating Select Boxes, Creating Radio Buttons, Creating Checkboxes, Creating Email, Special Field Types, The <meter> Element, Validating Forms.	9
Module 4: CSS Fundamentals and Selectors What Are Styles? What Does Cascading Mean? CSS Syntax, Using CSS on Your Webpage, External Style Sheets, Commenting Your CSS Code, Targeting Elements by Tag, Targeting Elements by Class, The Cascade, Inheritance, and Parent-Child, Relationships, Selecting Elements by their Relationships, Specificity and Precedence in the Cascade, Targeting Elements with Specific Attributes, Advanced Targeting.	9
Module 5: CSS Styling and Page Layout Choosing Fonts, Google Fonts, Including External Fonts with @font-face, Sizing Text, Formatting Text, How Computer Monitors Work Representing Color in CSS, Gradients, The border Property, The Box Model Padding and Margins, Element Flow, Creating Layers and Overlapping Elements, Creating an Overlay Using z-index.	9

Text Books:

1. Joe Casabona, Visual Quick-start Guide, HTML and CSS, 9th Edition, Peachpit Press, 2020.

Reference Books:

1. Jon Duckett, HTML and CSS: Design and build Websites, Wiley, 2019.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	**% Distribution**
Performance	30
Quiz	10
Total	40

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Code	**Course Delivery Method**
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects

Code	Course Delivery Method
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	1	2
CO2	3	2	3	3	2	0	1	2
CO3	2	3	3	3	2	0	1	2
CO4	2	2	3	3	1	0	1	2
CO5	3	3	3	3	3	1	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD6, CD7
CO2	CD1, CD2, CD4, CD5, CD6, CD7
CO3	CD1, CD2, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26511

Course Title: Operating Systems

Pre-requisite(s): Data Structures

Co-requisite(s): Python in Linux

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. Sandip Dutta, Dr. Jit Mukherjee

COURSE OBJECTIVES

This course enables the students to:

1. Present the main components of OS and their working
2. Introduce the concepts of process and thread and their scheduling policies
3. Introduce the various memory management techniques.
4. Analyze the different techniques for managing memory, I/O, disk and files.
5. Introduce the security and protection features of an Operating System.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Describe the main components of OS and their working.
CO2	Explain the concepts of process and thread and their scheduling policies.
CO3	Explain the various memory management techniques.
CO4	Compare the different techniques for managing memory, I/O, disk and files.
CO5	Explains the security and protection features of an Operating System.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Overview of Operating Systems and Processes: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Personal Computer, Parallel, Distributed Systems, Real- Time Systems, Special - Purpose Systems, Operating System services, user OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Modern Operating Systems and its Architecture. Process concepts - The Process, Process State, Process Control Block	**8**
Module 2: Threads and Scheduling Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multi-level Queue Scheduling, Multi-level Feedback Queue Scheduling, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Performance Analysis of Scheduling Policies, Case studies: Linux, Windows.	**8**
Module 3: Process Coordination and File Systems: Process Synchronization, The Critical section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors. File System and IOCS, Files and File Operations, Fundamental File Organizations, Directory Structures, File Protection, Mounting Interface between File System and IOCS, Allocation of Disk Space, Implementing File Access, File Sharing Semantics, File System Reliability, Virtual File System, Unix File System, Linux File System, Windows File System, Implementation, Efficiency and Performance of File Systems.	**8**
Module 4: Memory Management and Virtual Memory Managing the Memory Hierarchy, Static and Dynamic Memory Allocation, Memory Allocation to a Process, Reuse of Memory, Contiguous Memory Allocation, Noncontiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Kernel Memory Allocation, Allocation of Frames, A Review of Relocation, Linking and Program Forms, Thrashing. Virtual Memory Basics, Demand Paging, Page Replacement Policies, Memory Allocation to a Process, Shared Pages, Memory Mapped Files, Unix Virtual Memory, Linux Virtual Memory, Virtual Memory using Segmentation.	**8**
Module 5: Deadlocks, Security and Protection System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. Overview of Security and Protection, Goals of Security and Protection, Security Attacks, Formal and Practical aspects of Security, Encryption, Authentication and Password Security, Access Descriptors and the Access Control Matrix,	**8**

Protection Structures, Capabilities, Unix Security, Linux Security, Windows Security.	

Text Books:

1. Dhamdhere D.M., Operating Systems: A Concept-Based Approach, 2nd Edition, TMH, New Delhi, 2006.

Reference Books:

1. Silberschatz A., Galvin Peter B., Greg Gagne, “Operating System Concepts”, 6th Edition, John Wiley, Indian Reprint, 2003.
2. Crowley C., “Operating Systems: A Design-Oriented Approach”, TMH, New Delhi, 2002.
3. Deitel H.M., “Operating Systems”, 2nd Edition, Pearson Education, 2003.
4. Tanenbaum A.S., “Operating System: Design and Implementation”, PHI, New Delhi, 2002.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

<u>Direct Assessment</u>

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	20
Assignment(s)	10
Seminar before a committee	20
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Teaching aids/Seminars

Code	Course Delivery Method
CD4	Mini Projects
CD5	Expert talks
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	3	2	3	1	3
CO2	3	2	2	3	2	3	1	3
CO3	3	2	2	3	3	3	1	3
CO4	3	3	3	3	3	3	1	3
CO5	3	3	3	3	3	3	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD3
CO2	CD1, CD2, CD3, CD6
CO3	CD1, CD2, CD3, CD6
CO4	CD1, CD3, CD4, CD6
CO5	CD1, CD4, CD6

COURSE INFORMATION SHEET

Course Code: CA26513

Course Title: Analysis of Algorithms

Pre-requisite(s): Data Structures

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. Srabani Mukhopadhyaya, Dr. Ritesh Jha

COURSE OBJECTIVES

This course enables the students to:

1. Equip with the knowledge of fundamental algorithm design techniques and paradigms.
2. Equips with the basic concepts and methods used to analyze the efficiency and performance of algorithms
3. Assess and contrast different algorithms addressing the same problem.
4. Familiarize with the challenges of solving computationally complex, intractable problems and the limitations of algorithmic problem-solving.
5. Equip with the skills to formulate efficient algorithms for common and real-life problems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Determine the asymptotic complexity of an algorithm including solving of recurrence relations.
CO2	Understand the basic paradigms used in algorithm design, like divide and conquer, greedy, dynamic programming, backtracking, and branch & bound.
CO3	Analyze and compare different algorithms for a given problem based on their correctness, efficiency, and resource utilization.
CO4	Comprehend NP-complete problems and the inherent limitations of algorithmic power and computational tractability.
CO5	Develop efficient algorithms for typical design situations and practical problems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction, Asymptotic Notations, Analysis of recursive algorithms Characteristics of algorithms, Performance measurements of algorithms - best, average and worst-case behavior, Time and space trade-offs. Growth of functions, Asymptotic notations: O, θ, Ω, o, ω Operations on asymptotic notations. Methods to solve recurrences - substitution method, recursion tree method, Master's Theorem: Statement and applications.	8
Module 2: Divide-and-conquer Basic ideas, Binary search, Merge sort, Quick sort, Randomized quick sort, Finding the median and selection problem, Strassen's matrix multiplication with Space and time complexity analysis.	8
Module 3: Greedy Algorithms General characteristics of greedy algorithms, Knapsack problem, Minimum cost spanning tree problem: Prim and Kruskal algorithms, Shortest path problem: Dijkstra and Bellman-Ford algorithms, emphasizing implementation issues and complexity analysis.	8
Module 4: Dynamic Programming Introduction and basic concepts using simple examples, like, calculation of Fibonacci numbers, calculation the binomial coefficients, etc.; 0-1 Knapsack problem, All pair shortest paths problem (Floyd-Warshall algorithm), Finding transitive closure, Matrix chain multiplication, Travelling salesperson problem with Space and time complexity analysis included.	8
Module 5: Advanced Algorithm Design and Computational Complexity Graph Traversal: Depth First Search and Breadth First Search Limitations of Algorithmic Power: Backtracking: n-Queen problem, Sum of subsets problem Branch and Bound: 0/1 Knapsack problem, Assignment problem Computational Intractability: Non-deterministic algorithms (sorting, searching, satisfiability, 0/1 Knapsack problem), P vs NP, Classes NP-hard and NP-Complete, Reducibility and Statement of Cook's Theorem.	8

Text Books:

1. Brassard G. & Bratley P., Fundamentals of Algorithms, Pearson and Prentice Hall of India, New Delhi, 2005.

2. Horowitz Ellis, Sahni Sartaj and Rajasekaran Sanguthevar, Fundamentals of Computer Algorithms, University Press (I) Pvt. Ltd., 2012.

Reference Books:

1. Cormen Thomas H., Leiserson Charles E., Rivest Ronald L. and Stein Clifford, Introduction to Algorithms, PHI publication, 3rd Edition, 2009.
2. Levitin Anany, Introduction to Design and Analysis of Algorithms, 3rd Edition, Pearson, 2012
3. Aho Alfred V., Hopcroft John E. & Ullman Jeffrey D., The Design & Analysis of Computer Algorithms, Addison Wesley Publications, Boston, 1983.
4. Kleinberg Jon & Tardos Eva, Algorithm Design, Pearson Education, 2006.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	1	2	0	0
CO2	3	2	2	2	1	2	0	0
CO3	3	3	3	3	2	3	1	1
CO4	3	3	2	3	1	1	0	0
CO5	3	3	3	3	2	3	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5, CD6, CD7
CO2	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO3	CD1, CD2, CD4, CD5, CD6
CO4	CD1, CD2, CD3, CD7
CO5	CD4, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26515

Course Title: Python Programming

Pre-requisite(s): Basic Programming Concepts

Co-requisite(s): Python in Linux

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Prof. Abhijit Mustafi, Dr. Shruti Garg

COURSE OBJECTIVES

This course enables the students to:

1. Understand Python syntax, data types, operators, and fundamental programming constructs.
2. Develop modular Python programs using functions, modules, and packages.
3. Apply Python data structures to solve computational problems efficiently.
4. Understand file handling, exception handling, and object-oriented programming concepts in Python.
5. Develop Python applications following good programming practices and coding standards.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain Python syntax, data types, operators, and control structures.
CO2	Develop modular Python programs using functions, modules, and packages.
CO3	Apply Python data structures and built-in libraries to solve computational problems.
CO4	Implement file handling and exception handling techniques in Python applications.
CO5	Design and develop object-oriented Python applications using appropriate programming practices.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Python Fundamentals Basic data types, literals and variables in Python, built in data types, objects, basic input output, operators and expressions, precedence rules.	6
Module 2: Control Flow and Functions Control flow, writing functions in Python, scope of variables, function arguments, anonymous functions, library functions and modules, packages.	6
Module 3: Collections and Data Structures String handling, lists and tuples in Python, list comprehensions, built in methods for tuples, arrays, sets and dictionaries, frozenset and set comprehensions, built in methods for dictionaries and their usage.	10
Module 4: File Handling and Exception Management Handling files, basic operations, memory management, working with CSV files, simple directory handling, errors and exception handling, Assertions.	6
Module 5: Classes and Object-Oriented Concepts Object oriented programming in Python, encapsulation, private and public members, methods, built in attributes and functions, inheritance, abstract and metaclass, polymorphism and operator overloading.	12

Text Books:

1. Anita Goel, Python Programming: An Object Oriented Approach, Universities Press, 2025

Reference Books:

1. Al Sweigart, Automate the Boring Stuff with Python, No Starch Press, 3rd Edition., 2025

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

<u>Direct Assessment</u>

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Lecture notes
CD3	Reading assignments
CD4	Programming assignments for implementation
CD5	Seminars
CD6	Small group projects
CD7	Self- learning such as use of NPTEL materials and internets

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	1	2
CO2	3	3	2	2	1	0	1	2
CO3	2	3	3	2	1	0	1	2
CO4	2	2	2	3	1	0	2	2
CO5	2	3	3	3	2	1	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD5
CO4	CD1, CD5, CD6
CO5	CD1, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26517

Course Title: Software Design Patterns

Pre-requisite(s): Understanding OOP using Java

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. K. Sridhar Patnaik, Dr. S. P. Singh

COURSE OBJECTIVES

This course enables the students to:

1. Understand the need for software design patterns and object-oriented design principles.
2. Apply creational, structural, and behavioral patterns in software development.
3. Analyze software design problems and select appropriate design patterns.
4. Develop maintainable and scalable software solutions using industry-standard practices.
5. Evaluate and optimize software architectures using design patterns and best practices

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain software design principles, object-oriented design concepts, and the role of design patterns in software development. *(Bloom's Level: Understand – L2)*
CO2	Apply creational, structural, and behavioral design patterns to solve recurring software design problems. *(Bloom's Level: Apply – L3)*
CO3	Analyze software requirements and identify suitable design patterns for improving maintainability, flexibility, and scalability. *(Bloom's Level: Analyze – L4)*
CO4	Evaluate alternative design solutions and justify the selection of appropriate design patterns for enterprise applications. *(Bloom's Level: Evaluate – L5)*
CO5	Design and Develop industry-standard software systems incorporating modern architectural and design patterns to address real-world problems. *(Bloom's Level: Create – L6)*

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Fundamentals of Software Design and Design Patterns Introduction to Software Design, Software Quality Attributes: Maintainability, Reusability, Scalability, Extensibility Object-Oriented Analysis and Design (OOAD), SOLID Principles, GRASP Principles, DRY, KISS, YAGNI Principles, Introduction to Design Patterns, Classification of Design Patterns, UML for Design Patterns, Anti-Patterns and Code Smells.	9
Module 2: Creational Design Patterns Factory Method Pattern, Abstract Factory Pattern, Builder Pattern, Prototype Pattern, Singleton Pattern, Dependency Injection Concept, Object Creation Best Practices, Pattern Selection Guidelines, Case Studies: Database Connection Management, Configuration Management Systems, Application Framework Initialization	9
Module 3: Structural Design Patterns Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Facade Pattern, Flyweight Pattern, Proxy Pattern, Layered Architecture Concepts, Design Pattern Integration. Case Studies: Payment Gateway Integration, Logging Frameworks, API Wrappers and Middleware Components.	9
Module 4: Behavioral Design Patterns Chain of Responsibility Pattern, Command Pattern, Interpreter Pattern, Iterator Pattern, Mediator Pattern, Memento Pattern, Observer Pattern, State Pattern, Strategy Pattern, Template Method Pattern, Visitor Pattern. Case Studies: Event-Driven Systems, Notification Services, Workflow Automation Systems, Chat Applications	9
Module 5: Enterprise and Modern Software Design Patterns MVC (Model-View-Controller), MVVM Pattern, Repository Pattern, Dependency Injection Pattern, Service Layer Pattern, DAO (Data Access Object) , Microservices Design Patterns : API Gateway, Circuit Breaker, Service Discovery, Saga Pattern , Event-Driven Architecture, Domain-Driven Design (DDD) Basics, Cloud-Native Design Considerations, Pattern Selection and Refactoring Techniques, Industry Case Studies: Netflix Architecture, Amazon Microservices, Spring Boot Applications, Cloud-Based Enterprise Systems	9

Text Books:

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesle, 1995.
2. Robert C. Martin, Agile Software Development: Principles, Patterns, and Practices, PHI, 2002.

Reference Books

1. Eric Freeman & Elisabeth Robson, Head First Design Patterns, O'Reilly, 2004.
2. Martin Fowler, Refactoring: Improving the Design of Existing Code, Addison-Wesley, 2019.
3. Robert C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, Pearson India, 2017.
4. Martin Fowler, Patterns of Enterprise Application Architecture, Addison-Wesle, 2002.
5. Chris Richardson, Microservices Patterns: With Examples in Java, Manning, 2018.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Interactive lectures, flipped classroom activities
CD2	Demonstration of GoF patterns, hands-on laboratory exercises, coding assignments, problem-solving sessions
CD3	Case-based learning, design analysis workshops, pattern identification exercises, group discussions
CD4	Interactive lectures, flipped classroom activities
CD5	Industry case studies, design reviews, comparative analysis of alternative solutions, seminar presentations
CD6	Mini-projects, enterprise application development
CD7	project-based learning using modern frameworks

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	0	2
CO2	3	2	3	2	1	0	0	1
CO3	2	3	3	2	1	0	0	1
CO4	1	3	2	2	1	1	1	2
CO5	2	3	3	3	2	1	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD2
CO3	CD1, CD2, CD3
CO4	CD4, CD5
CO5	CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26519

Course Title: Data Communication and Computer Networks

Pre-requisite(s): None

Co-requisite(s): Data Communication and Computer Networks Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. S. Kanungo, Dr. Rathindra Nath Dutta

COURSE OBJECTIVES

This course enables the students to:

1. Provide fundamental knowledge of data communication principles, computer network architectures, and communication models.
2. Understand various networking protocols, standards, transmission media, and network technologies.
3. Explain error control, flow control, routing techniques, and protocols used in different network layers.
4. Develop understanding of transport and application layer services including TCP, UDP, HTTP, DNS, and email protocols.
5. Understand modern networking technologies and network security concepts.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts of data communication, network models, topologies, protocols, and transmission techniques.
CO2	Analyze error detection, error correction, flow control, and medium access control mechanisms used in computer networks.
CO3	Apply IP addressing, subnetting, routing algorithms, and network layer protocols for designing and managing networks.
CO4	Describe transport and application layer protocols and analyze their role in reliable communication over networks.
CO5	Understand modern networking technologies, security mechanisms, wireless networks, IoT protocols, and emerging network architectures.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Fundamentals of Data Communication and Computer Networks Communication Model; Network Topologies; Transmission Modes; Standards and Protocols: OSI and TCP/IP Models (overview, functions and protocols in each layer), Comparison of the two models; Types of Networks (LAN, MAN, WAN, WLAN, PAN), Internet; Guided and Unguided Medium (overview of design and propagation mechanism); Overview of Modulation and Encoding Techniques (Digital-to-Digital, Analog-to-Digital); Overview of Multiplexing: FDM, TDM, STDM, WDM; Overview of Switching Techniques: Circuit Switching, Message Switching, and Packet Switching.	8
Module 2: Medium Access Control and Data Link Control Error Detection and Correction: Parity, Checksum, CRC, and Hamming Code; Flow and Error Control Protocols: Stop-and-Wait, Go-Back-N ARQ, Selective Repeat ARQ; Overview of High-Level Data Link Control (HDLC) and Point-to-Point Protocol (PPP) Protocols; Random Access Protocols: ALOHA, CSMA, CSMA/CD, and CSMA/CA; Controlled Access and Channelization (FDMA, TDMA, CDMA); List of IEEE 802 Standards.	8
Module 3: Network Layer IP Address: Overview, Classful Addressing and Subnetting, Classless Inter-Domain Routing (CIDR), IPv4 Header Format; Network Address Translation (NAT); IPv6: Architecture, Header Format, IPv4-to-IPv6 Transition Strategies; Overview of Network Routing: Static Routing, Flooding, Dynamic Routing, Link-State Routing, Distance-Vector Routing, OSPF and RIP protocols, Path Vector Routing (BGP); Connecting Devices (Switches, Bridges, Routers).	8
Module 4: Transport and Application Layer Transport Layer Services, Port Address, Connection-Oriented vs Connectionless Protocols; User Datagram Protocol (UDP): Header format and Reliability; Transmission Control Protocol (TCP): Header Format, Connection Establishment & Termination, Congestion Control in TCP; Quality of Service (QoS) Principles and Traffic Shaping (Leaky Bucket, Token Bucket); Client-Server and Peer-to-Peer Applications, Web and HTTP (Non-Persistent and Persistent Connections), Electronic Mail (SMTP, POP3, IMAP) and MIME Types, File Transfer (FTP); Domain Name System (DNS) and Address Resolution.	8
Module 5: Modern Networking Technologies Wireless LAN (IEEE 802.11), Bluetooth, Virtual Local Area Networks (VLAN); Overview of Video Streaming and Content Distribution Networks; Overview of Software-Defined Networking (SDN): Control Plane vs. Data Plane, OpenFlow Protocol, Network Function Virtualization (NFV); Overview of Internet of Things (IoT) Communication Protocols (MQTT, CoAP); Overview of IPsec, Virtual Private Networks (VPN), TLS and Firewalls.	8

Text Books:

1. James F. Kurose & Keith W. Ross, Computer Networking: A Top-Down Approach, Pearson, 7th Edition, 2016.
2. Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill, 5th Edition, 2017.
3. William Stallings, Data and Computer Communications, Pearson Education, 10th Edition, 2017.

Reference Books:

1. Andrew S. Tanenbaum & David J. Wetherall, Computer Networks, Pearson, 2010.
2. Larry L. Peterson & Bruce S. Davie, Computer Networks: A Systems Approach, Morgan Kaufmann, 6th Edition, 2021.
3. Douglas E. Comer, Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture, **Pearson Education India,** 6th Edition, 2015.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors

Code	Course Delivery Method
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	1	0	1	2
CO2	3	3	2	2	1	0	1	2
CO3	3	3	3	3	1	1	1	2
CO4	3	2	2	3	1	1	2	2
CO5	2	2	2	3	2	1	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4, CD5, CD6, CD7
CO2	CD1, CD2, CD3, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26516

Course Title: Python in Linux

Pre-requisite(s): None

Co-requisite(s): Python Programming, Operating Systems

Credits: 1.5 L: 0 T: 0 P: 03 SL:

Class Schedule per Week: 03

Semester / Level: II / V

Branch: Master of Computer Applications

Name of the Teacher: Prof. Abhijit Mustafi

COURSE OBJECTIVES

This course enables the students to:

1. Understand the Linux operating environment and basic command-line utilities for software development.
2. Develop Python programs using variables, control structures, functions, and modular programming concepts.
3. Apply Python data structures and file handling techniques to solve computational problems.
4. Design object-oriented Python applications using classes, objects, and constructors.
5. Analyze and visualize data using Python libraries such as NumPy and Matplotlib.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Demonstrate proficiency in the Linux operating environment and develop Python programs using fundamental programming constructs such as variables, operators, decision-making, and iterative statements. (BL3)
CO2	Develop modular Python programs using functions, parameter passing, and built-in data structures including lists, tuples, dictionaries, and sets. (BL3)
CO3	Apply file handling techniques and standard Python modules to develop efficient solutions for computational problems. (BL3)
CO4	Design and implement object-oriented Python applications using classes, objects, constructors, and inheritance principles. (BL6)
CO5	Analyze, process, and visualize data using NumPy and Matplotlib to solve real-world computational and data analysis problems. (BL4)

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundation of Python in Linux Introduction to basic Linux commands, directory traversal, file manipulation. Basic Python programming, variables, selection structures, iterations.	9
Module 2: Python Data Structures and Functions Basic list handling. formatted input output, writing functions, argument passing	6
Module 3: Python Collections and File Handling Working with lists – list comprehensions, nested list structures, dictionaries, tuples and sets, file handling, some common modules including os, itertools, collections etc.	9
Module 4: Classes and Objects in Python Object oriented python programming, building classes, writing constructors, instantiating classes.	6
Module 5: Python Data Processing and Visualization Basic data handling in Python, Introduction to numpy, data visualization with Matplotlib	9

Text Books:

1. E. Matthess, Python Crash Course, No Starch Press, 2023.

Reference Books:

1. A. Sweigart, Automate the Boring Stuff with Python, No Starch Press, 2026.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	1	0	1	2
CO2	3	3	2	3	2	0	1	2
CO3	3	3	3	2	2	0	1	2
CO4	2	3	3	3	2	1	1	3
CO5	2	3	3	3	2	1	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD5, CD6, CD7
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD4, CD5, CD6
CO4	CD1, CD2, CD4, CD5, CD6
CO5	CD1, CD2, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26520

Course Title: Data Communication and Computer Networks Lab

Pre-requisite(s): Data Structures using Java

Co-requisite(s): Data Communication and Computer Networks, Operating Systems, Python in Linux

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. Rathindra Nath Dutta, Dr. Prashant Pranav

COURSE OBJECTIVES

This course enables the students to:

1. To introduce students to fundamental networking concepts, architectures, and types of networks.
2. To develop understanding of networking devices, their configurations and different protocols.
3. To provide hands-on experience with networking tools for network analysis, monitoring, and troubleshooting.
4. To build conceptual and practical knowledge of IP addressing, subnetting, socket programming, and client-server communication.
5. To enable students to analyze, simulate, and evaluate network protocols, traffic behavior, and routing algorithms using tools and programming.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain and design network topologies and architectures.
CO2	Configure and troubleshoot networking devices and interfaces using standard networking tools.
CO3	Analyze and manage IP addressing, subnetting and routing schemes.
CO4	Develop basic network applications using socket programming.
CO5	Evaluate network performance and behavior using tools and simulations.

SYLLABUS

<table>
<tr><th>MODULE</th><th>NO. OF LECTURE HOURS</th></tr>
<tr><td>

Module 1: Understand basic networking concepts, devices, and tools

1. Design of various network topologies
2. Identify and configuration of networking Devices: hub, switch, router, gateway
3. Design of LAN, WLAN, MAN, WAN
4. Using various transmission modes

Use of network tools in Unix-like systems: ip, ifconfig, ping, traceroute, nslookup, netstat, nmap, iperf, etc.

5. Display all network interfaces and configuration (IP address, Subnet mask, gateway, MAC address etc.); enable/disable interfaces
6. Test connectivity to local devices and public servers; observe packet loss and latency.
7. Compare routes to different destinations; observe number of hops, delay at each hop.
8. Obtain domain information of common websites observe response time.
9. Identify open/listening ports in local machine; identify which processes are using a given port.
10. Scan your network (small subnet) to identify active devices, open ports, running services etc.
11. Test network performance in UDP and TCP connections; identify bandwidth, packet loss, jitter etc.

</td><td>6</td></tr>
<tr><td>

Module 2: Addressing and Socket Programming

Basic understanding of physical address and logical addresses

12. Identify MAC Address and OUI; classification of MAC address into unicast/broadcast/multicast addresses.
13. Represent IPv4 address in different notations.
14. Classify given IPv4 addresses into class A/B/C/D/E.
15. Given an IPv4 address in CIDR notation find the subnet mask, network ID, broadcast address and host range.

Basic understanding of ports, socket address, socket programming design flow

</td><td>9</td></tr>
</table>

16. Design an echo server with socket programming and communicate using a client (Telnet) 17. Designing custom client-server program for data processing at the server 18. Efficiently handling multiple clients: a chat application.	
Module 3: Basic data communication operations 19. Implement bit-stuffing and byte-stuffing. 20. Implement various digital encoding techniques. 21. Implement various error detection schemes: VRC, LRC, CRC, checksum etc. 22. Implement Hamming code for error correction. 23. Implement simulations of various events in Stop-and-Wait ARQ and Sliding Window Protocols. 24. Examine routing tables. 25. Implement Dijkstra and Bellman-Ford Algorithms to find shortest routes between two nodes. 26. Find broadcast tree in a given network.	9
Module 4: Understanding network traffic Packet Analysis using tcpdump/Wireshark and observe different fields: 27. Capturing and analyzing Ethernet/WLAN frames. 28. Examine ARP/RARP packets; observe the broadcast/unicast nature of the request/reply. 29. Analyze TCP connections; observe 3-way handshake, 4-way handshake, half-close. 30. Analyzing Packets for DNS Lookup, HTTP Request-Response. 31. Extraction data from un-encrypted HTTP traffic. 32. Implement a simplified PING utility: measure RTT. average packet loss and latency 33. Implement basic Traceroute simulation: show hop count and delay.	9
Module 5: Network Simulation (ns-3 / Cisco Packet Tracer / NetSim / Mininet) 34. Design various topologies with routers and switches. 35. Create subnets and allocate IPs (static and dynamic assignment). 36. Configure static routing manually; observe routing tables and connectivity.	6

37. Apply dynamic routing strategies; observe routing table updates. 38. Testing connectivity/bandwidth between nodes. 39. Simulation of different types of traffic. 40. Familiarities with advanced topics: congestion control, VLAN, User Plane/Control Plane, Security etc.	

Text Books:

1. James F. Kurose & Keith W. Ross, Computer Networking: A Top-Down Approach, Pearson Education, 7th Edition, 2016.
2. Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill Education, 5th Edition, 2017.

Reference Books:

1. Andrew S. Tanenbaum & David J. Wetherall, Computer Networks, Pearson Education, 5th Edition, 2010.
2. William Stallings, Data and Computer Communications, Pearson Education, 10th Edition, 2017.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	20
Total	60

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	2	1	1	1
CO2	2	2	3	2	3	2	1	2
CO3	3	3	2	3	3	1	1	1
CO4	2	2	3	2	3	2	1	2
CO5	3	3	2	3	3	1	1	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4, CD5
CO2	CD1, CD4, CD5
CO3	CD1, CD2, CD3, CD5, CD6, CD7
CO4	CD1, CD2, CD4, CD5
CO5	CD1, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26522

Course Title: Learning Power BI Applications

Pre-requisite(s): Database Management System

Co-requisite(s): None

Credits: 02 L: 0 T: 0 P: 02

Class Schedule per Week: 04

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. Debjani Mustafi, Dr. Komal Naaz

COURSE OBJECTIVES

This course enables the students to:

1. To introduce students to Business Intelligence concepts and the Power BI Desktop environment for data-driven decision making.
2. To develop practical skills in importing, cleaning, transforming, and managing datasets using Power BI tools.
3. To enable students to create interactive visualizations and dashboards for effective data presentation.
4. To provide hands-on experience in applying data modeling and basic DAX functions for analytical reporting.
5. To develop the ability to design and present dashboard-based solutions using the Power BI Desktop.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Import, clean, and transform data from multiple sources using Power BI Desktop for analytical reporting.
CO2	Create and customize visualizations and interactive dashboards to represent data effectively.
CO3	Apply data modeling concepts and DAX functions to generate analytical insights.
CO4	Design business-oriented reports and perform exploratory analysis using Power BI tools.
CO5	Develop and present an end-to-end dashboard-based mini project using the free version of Power BI Desktop.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Power BI Desktop Introduction to Business Intelligence and Power BI environment, importing datasets from Excel, CSV and databases, Data cleaning and preprocessing, Handling missing and duplicate values.	4
Module 2: Introduction to Power Query Editor Data transformation using Power Query, Merge and append operations, Data profiling and preparation.	6
Module 3: Data Visualization and Dashboard Development Creating charts and visual reports, Applying filters, slicers, and drill-down analysis, Dashboard design principles, Interactive dashboard creation, Report formatting and customization.	6
Module 4: Advanced Reporting and Analytical Applications Creating calculated columns and measures using DAX, Time-based and comparative analysis, Key Performance Indicator (KPI) development and analytical storytelling, to create multi-page interactive reports, Case studies using educational, finance, HR, or sales datasets.	6
Module 5: Dashboard Design and Mini Project Dashboard layout and usability principles, Report optimization and formatting, Exporting reports (PDF/PPT options available in Desktop) Mini Project: End-to-end dashboard development using local datasets, Project presentation and demonstration Sample Projects: Student Performance Dashboard, Admission Analytics Dashboard, Retail Sales Dashboard, HR Attrition Dashboard, Public Dataset Analytics	8

Text Books:

1. Dan Clark, Beginning Microsoft Power BI, Apress, CA Publication, ISBN: 978-1-4842-5620-6, 2020.
2. Alberto Ferrari and Marco Russo, Introducing Microsoft Power BI, Microsoft Press, ISBN: 978-1-5093-0228-4, 2016.

Reference Books:

1. (Online) Microsoft Power BI Documentation (Power BI documentation - Power BI | Microsoft Learn)
2. Reza Rad, Power BI from Rookie to Rock Star – Book one: Power BI Essentials, RADACAD Systems Limited, 2019.
3. Brett Powell, Microsoft Power BI Cookbook, Packt Publications, ISBN: 9781788290142, 2017.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments

Code	Course Delivery Method
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	1	2	3
CO2	3	3	3	3	3	1	2	3
CO3	3	3	3	3	3	1	2	3
CO4	3	3	3	3	3	1	2	3
CO5	3	3	3	3	3	1	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4, CD5, CD7
CO2	CD1, CD2, CD4, CD5, CD6, CD7
CO3	CD1, CD2, CD4, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD7
CO5	CD1, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26601

Course Title: Machine Learning

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): Machine Language Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Vandana Bhattacharjee, Dr Komal Naaz, Dr Niraj Kumar Singh

COURSE OBJECTIVES

This course enables the students:

1.	To understand the Mathematical Foundations of machine learning.
2.	To learn about various supervised machine learning algorithms.
3.	To get a grasp over the semi-supervised and unsupervised machine learning algorithms.
4.	To understand the theory related to the evaluation and validation of machine learning models.
5.	To understand the working of neural networks for building real life applications by deep learning.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Apply the basics of linear algebra matrices, probability theory and problem formulation.
CO2	Demonstrate understanding of a range of supervised learning mechanisms.
CO3	Demonstrate understanding of the unsupervised and semi-supervised learning mechanisms.
CO4	Apply the various model evaluation and validation techniques.
CO5	Develop an understanding of the neural networks in the context of deep learning and the role of backpropagation in training the networks.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Machine Learning Definition of Machine Learning, types of learning (Supervised, Unsupervised, Reinforcement). Revisiting probability, linear algebra matrices. Data Preprocessing: Handling missing values, data normalization, scaling, and categorical encoding. Principal Component Analysis (PCA) for reducing feature spaces.	8
Module 2: Supervised Learning Linear Regression, and Polynomial Regression. Logistic Regression, K-Nearest Neighbours (KNN), Support Vector Machines (SVM), and Naive Bayes. Decision Trees, Information Gain, Entropy, and Random Forest ensemble methods.	8
Module 3: Unsupervised and Semi Supervised Learning Clustering Techniques: K-Means Clustering, Hierarchical Clustering, and Density-Based Spatial Clustering (DBSCAN). Expectation maximization (EM) for soft clustering, Semi-supervised learning with EM using labeled and unlabeled data.	8
Module 4: Model Evaluation Performance Metrics: Confusion Matrix, Accuracy, Precision, Recall, F1-Score, and ROC-AUC curves. Overfitting & Underfitting: Bias-Variance tradeoff, regularization techniques (Lasso and Ridge regression). Validation Techniques: Train-test split, K-Fold Cross-Validation.	8
Module 5: Introduction to Deep Learning & Applications Neural Network Basics: Perceptron, Multi-Layer Perceptron (MLP), and activation functions (ReLU, Sigmoid). Gradient Descent and Perceptron/Delta Learning, Backpropagation and Loss functions. ML Application Domains: Basic text analysis (NLP) and introductory computer vision implementation.	8

Text Books:

1. Marsland S., Machine Learning: An Algorithmic Perspective, CRC Press, 2nd Edition, 2015
2. Han J., Pei J., Tong H., Data Mining: Concepts and Techniques, Morgan Kaufmann, 4th Edition, 2022

Reference Books:

1. Mitchell T. M., Machine Learning, McGraw-Hill, 1st Edition, 1997
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Teaching Aid (Document Material)
CD6	Online Material
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	0	1	1
CO2	3	3	3	3	2	2	1	2
CO3	3	3	3	3	2	2	1	2
CO4	3	3	3	3	2	1	1	2
CO5	3	3	3	3	2	2	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5
CO2	CD1, CD2, CD3, CD4, CD5
CO3	CD1, CD2, CD3, CD4, CD5
CO4	CD1, CD2, CD3, CD4, CD5, CD6
CO5	CD1, CD3, CD4, CD5, CD6

<u>**COURSE INFORMATION SHEET**</u>

Course Code: CA26603

Course Title: Cryptography

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Sandip Dutta, Dr. Amrita Priyam, Dr. Prashant Pranav

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts of cryptography, network security, security services, security mechanisms, and classical cryptographic techniques.
2. Develop a strong mathematical foundation in modular arithmetic, finite fields, and number theory required for the design and analysis of cryptographic algorithms.
3. Understand the principles, design, and implementation of symmetric-key cryptographic algorithms and key management techniques for secure communication.
4. Analyze asymmetric-key cryptographic algorithms, digital signatures, key exchange mechanisms, and Public Key Infrastructure (PKI) for authentication and confidentiality.
5. Apply cryptographic hash functions, message authentication techniques, and secure communication protocols to address practical network security challenges.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand the principles of cryptography and secure communication.
CO2	Apply symmetric and asymmetric encryption algorithms.
CO3	Analyze authentication mechanisms, digital signatures, and PKI.
CO4	Evaluate network security protocols and secure communication systems.
CO5	Identify common network attacks and implement appropriate security solutions.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Cryptography and Network Security Introduction to information security, security goals and services, security attacks and security mechanisms, security model for network security, cryptography and cryptanalysis, classical encryption techniques including substitution ciphers, transposition ciphers, product ciphers, steganography, and an overview of modern cryptographic systems and applications.	8
Module 2: Mathematical Foundations of Cryptography Integer arithmetic, divisibility and prime numbers, greatest common divisor, Euclidean and extended Euclidean algorithms, modular arithmetic, congruence, Euler's phi function, Fermat's theorem, Euler's theorem, finite fields, Galois fields, polynomial arithmetic, and mathematical foundations for cryptographic algorithms.	8
Module 3: Symmetric-Key Cryptography Principles of symmetric-key cryptography, stream ciphers and block ciphers, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), block cipher modes of operation, key distribution, key management, random number generation, and practical applications of symmetric encryption.	8
Module 4: Asymmetric-Key Cryptography Principles of public-key cryptography, RSA algorithm, Diffie–Hellman key exchange, ElGamal cryptosystem, elliptic curve cryptography (ECC), public key distribution, Public Key Infrastructure (PKI), X.509 certificates, digital signatures, key management, and applications of asymmetric cryptography.	8
Module 5: Cryptographic Hash Functions and Authentication Cryptographic hash functions, properties and applications of hash functions, SHA family of hash algorithms, message authentication, Message Authentication Codes (MAC), HMAC, digital signatures using hash functions, authentication protocols, user authentication, Secure Socket Layer (SSL), Transport Layer Security (TLS), and secure electronic mail using PGP and S/MIME.	8

Text Books:

1. Behrouz A. Forouzan, Introduction to Cryptography and Network Security, McGraw-Hill Education, 2007.

Reference Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, 8^{th} edition, Pearson, 2020.

2. Charlie Kaufman, Radia Perlman, Mike Speciner and Ray Perlner,3rd edition Network Security: Private Communication in a Public World, Pearson, 2024.

3. Atul Kahate, Cryptography and Network Security, 5th edition McGraw-Hill, 2026.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Interactive classroom lectures using presentations and board work
CD2	Tutorial sessions for mathematical problem solving and cryptographic algorithm analysis
CD3	Demonstration of cryptographic algorithms, security protocols, and network security tools
CD4	Assignment-based learning involving analytical and numerical problems
CD5	Case studies on cryptographic applications and network security incidents
CD6	Student seminars and presentations on emerging trends in cryptography and cybersecurity
CD7	Self-learning through textbook reading, research articles, and online learning resources

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	1	2
CO2	3	3	2	2	0	0	0	2
CO3	2	3	3	3	1	0	2	2
CO4	2	3	3	3	1	1	3	2
CO5	2	3	2	3	1	0	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD2, CD4, CD7
CO3	CD1, CD2, CD5, CD6
CO4	CD6, CD7
CO5	CD1, CD2, CD5, CD6

<u>**COURSE INFORMATION SHEET**</u>

Course Code: CA26602

Course Title: Machine Learning Lab

Pre-requisite(s): Mathematics for Computing, Python Programming

Co- requisite(s): - Machine Learning

Credits: 1.5 L: 0 T: 0 P: 03

Class schedule per week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of Teacher: Dr Vandana Bhattacharjee, Dr Nandini Kumari

COURSE OBJECTIVES

This course enables the students to:

1. Develop Python programs to implement fundamental machine learning algorithms, including Linear Regression and K-Nearest Neighbors, and evaluate model performance using standard metrics.
2. Implement and analyze supervised learning techniques such as Logistic Regression, Support Vector Machines, and Decision Trees for classification tasks.
3. Develop neural network models using Multilayer Perceptrons (MLPs) and the Backpropagation algorithm to solve machine learning problems.
4. Apply unsupervised learning techniques, including K-Means and Hierarchical Clustering, to identify patterns and group unlabeled data.
5. Implement ensemble learning methods such as Bagging and Boosting, and integrate machine learning concepts to develop a mini-project using a real-world dataset.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Preprocess and visualize data for machine learning tasks
CO2	Understand and implement basic supervised learning algorithms
CO3	Explore dimensionality reduction techniques

CO4	Design and apply unsupervised learning techniques
CO5	Analyze performance metrics of supervised and unsupervised models

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Machine Learning Programs based on Linear Regression, Gradient Descent, and the implementation of different evaluation metrics like MSE, RMSE, MAE, and R^2 Score. Programs on the K-Nearest Neighbor technique.	6
Module 2: Supervised Learning Programs based on Logistic Regression. Programs on Support Vector Machines (SVMs) using linear and different kernel approaches. Programs based on implementing Decision Trees.	9
Module 3: Neural Networks Programs based on understanding the concept of linear separability. Programs based on Multilayer Perceptron (MLP) and Back Propagation Algorithm.	9
Module 4: Unsupervised Learning Programs based on K-Means Clustering. Programs based on Hierarchical clustering methods	6
Module 5: Ensemble learning Programs based on Bagging and Boosting. Mini project on real data set.	9

Text Books:

1. Aurelien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media Inc., ISBN: 9781492032649, 2019.

Reference Books:

1. Tom M. Mitchell, Machine Learning, Tata McGraw Hill, New Delhi, 2017.

2. Shalev-Shwartz Shai, Ben-David Shai, Understanding Machine Learning, Cambridge University Press, 2017.

3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2009.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids

Code	Course Delivery Method
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	1	3	1	1	1	3
CO2	3	3	3	3	2	1	1	3
CO3	3	3	2	3	1	0	0	3
CO4	3	3	3	3	2	0	0	3
CO5	3	2	1	3	1	0	0	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD3, CD5
CO2	CD1, CD3, CD5
CO3	CD1, CD3, CD5
CO4	CD1, CD3, CD5, CD4
CO5	CD1, CD3, CD5, CD4

PROGRAM ELECTIVE I

TRACK: CORE COURSES

COURSE INFORMATION SHEET

Course Code: CA26521

Course Title: Computer Organization and Architecture

Pre-requisite(s): Data Structures

Co-requisite(s): None

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr. Sudip Kumar Sahana, Dr. Umesh Prasad

COURSE OBJECTIVES

This course enables the students to:

1. Understand the concepts of building blocks of digital logic design Computer Architecture
2. Conceptualize the software and hardware relationship through Instruction Set architecture, addressing modes and instruction formats
3. Develop the logical ability to design hardware blocks and control signals for specific instructions and set of instructions.
4. Explore memory organization, requirements and performance
5. Impart the knowledge of I/O organization and concept of parallel processing

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain concepts of building blocks of Computer Architecture and performance
CO2	Explore the software and hardware relationship through Instruction Set architecture, addressing modes and instruction formats
CO3	Design the hardware requirements for control signals for specific instructions, set of instructions and pipelining concept
CO4	Evaluate the memory organization, I/O organization and its impact on computer cost/performance
CO5	Analyze the advancement of Computer architecture using networking and parallel concepts

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Digital Logic and Computer Basics Boolean algebra, simplification of Boolean expressions, canonical forms. Karnaugh maps up to 4 variables, Combinational circuits: half adder, full adder, encoder, decoder, multiplexer, parity generator/checker. Memory basics: RAM and ROM, memory organization. Sequential circuits: latches, flip-flops, RS, JK, D, and master-slave flip-flops	8
Module 2: Instruction Set Architecture Memory locations and addresses, byte addressability. Big-endian and little-endian representation, Word alignment, Instructions and instruction sequencing, Addressing modes, Assembly language basics, Subroutines and procedure call basics, Immediate values and instruction formats	8
Module 3: Basic Processing Unit and Pipelining Basic processing unit, instruction execution cycle, Hardware components and instruction fetch/execute steps, Control signals and hardwired control, Pipelining concept, pipeline organization, hazards, data dependencies, branch delays, Pipeline performance evaluation	8
Module 4: Memory Organization Semiconductor RAM and ROM, Direct memory access, Memory hierarchy, Cache memory design and performance, Virtual memory, Memory management requirements	8
Module 5: Input Output and Parallel Concepts I/O basics, accessing I/O devices, interrupts, Bus structure, bus operation, arbitration, interfaces, interconnection standards. Parallel processing basics, Flynn's Taxonomy, SIMD/vector processing, Shared-memory multiprocessors, Cache coherence	8

Text Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw-Hill, 2023

Reference Books:

1. David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 6th Edition, Elsevier/Morgan Kaufmann, 2020.
2. M. Morris Mano, Rajib Mall, Computer System Architecture, 3rd revised edition, Pearson, 2024

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	2	1	0	1
CO2	3	3	3	2	2	1	0	1
CO3	3	3	3	3	2	1	0	1

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO4	3	3	3	3	2	1	0	1
CO5	3	3	3	3	2	1	0	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6
CO2	CD1, CD2, CD3, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26523

Course Title: Multimedia Technologies

Pre-requisite(s): None

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr Monu Bhagat, Dr Mansi Gupta

COURSE OBJECTIVES

This course enables the students:

1. Understand the fundamental concepts, components, architecture, and applications of multimedia systems.
2. Develop knowledge of digital signal processing principles, image representation, and multimedia data transformation techniques.
3. Apply image, audio, and video compression techniques using appropriate coding methods and multimedia standards.
4. Analyze multimedia communication systems and evaluate the performance of multimedia compression and transmission techniques.
5. Acquire the skills required to design and implement multimedia solutions for real-world applications using contemporary multimedia technologies.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand multimedia systems, signal processing fundamentals, and image representation techniques
CO2	Apply image transformation, coding, and compression techniques for multimedia data.
CO3	Analyze lossy/lossless compression algorithms and image compression standards such as JPEG and JPEG2000.
CO4	Analyze video compression techniques, motion estimation, and international multimedia standards including H.261, H.263, H.264, MPEG-1, and MPEG-2.
CO5	Evaluate multimedia communication systems and audio/video compression techniques for real-world multimedia applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Multimedia Fundamentals Introduction to Multimedia, Characteristics and Components of Multimedia, Multimedia Applications, Multimedia Hardware and Software, Multimedia Systems, Multimedia Authoring and Authoring Tools, Multimedia Communication and Applications.	8
Module 2: Digital Image and Audio Fundamentals Digital Image Representation, Image Characteristics, Colour Models, Image File Formats, Fundamentals of Digital Audio, Sampling and Quantization, Audio Representation, Audio File Formats, MIDI and Digital Audio.	8
Module 3: Multimedia Compression and Image Coding Need for Compression, Lossless and Lossy Compression, Run-Length Encoding, Huffman Coding, Arithmetic Coding, Transform Coding, Discrete Cosine Transform, Wavelet Transform, JPEG and JPEG2000.	8
Module 4: Video Fundamentals and Compression Digital Video Representation, Video Characteristics and Formats, Temporal and Spatial Redundancy, DPCM, Motion Estimation and Motion Compensation, Principles of Video Compression, MPEG-1, MPEG-2, H.261, H.263 and H.264.	8
Module 5: Audio Compression and Multimedia Systems Principles of Audio Compression, Predictive and Transform Coding, MPEG Audio Coding, MP3, Multimedia Communication Systems, Multimedia Databases, Multimedia Synchronization, Multimedia Storage and Retrieval, Multimedia Applications.	8

Text Books:

1. Ranjan Parekh, Principles of Multimedia, McGraw-Hill Education, 2nd Edition, 2013.

Reference Books:

1. Prabhat K. Andleigh and Kiran Thakrar, Multimedia Systems Design, Pearson Education, 1st Edition, 2003.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture using board/LCD
CD2	Assignments
CD3	Tutorials/Seminars
CD4	Mini Projects
CD5	Industrial Visit/Case Study
CD6	Self-learning (NPTEL, Internet)

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	3	2	2
CO2	3	2	3	2	2	3	1	2
CO3	3	3	3	2	3	3	3	2
CO4	2	3	3	3	2	2	2	2
CO5	3	2	2	2	3	3	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD2, CD4
CO3	CD1, CD2, CD4, CD5
CO4	CD3, CD4, CD5, CD6
CO5	CD1, CD2, CD3, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26525

Course Title: Data Compression

Pre-requisite(s): Data Structures

Co-requisite(s): None

Credits: 03 L: 03 T:0 P:0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr Saikat Chakraborty, Dr Devesh Kumar Upadhya

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts of data compression and information theory.
2. Analyze and implement various lossless compression algorithms.
3. Comprehend lossy compression techniques and their application in multimedia.
4. Evaluate compression efficiency using different mathematical and statistical models.
5. Apply compression techniques to real-world data storage and transmission scenarios.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the theoretical foundations of data compression, including entropy and information theory.
CO2	Design and implement lossless compression algorithms such as Huffman and Arithmetic coding, etc.
CO3	Apply dictionary-based and context-based compression techniques like LZ77/LZ78 to textual and image data, etc.
CO4	Analyze lossy compression standards and their impact on data quality, specifically for multimedia.
CO5	Evaluate and select appropriate compression strategies based on application requirements for storage and bandwidth efficiency.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Source Coding Techniques Information theory, Shannon's source coding theorem, modeling, physical models, probability models, Markov models, uniquely decodable codes, prefix codes.	8
Module 2: Lossless Data Compression Techniques Huffman algorithm, minimum variance, adaptive Huffman, Golomb/Rice codes, arithmetic coding, generating tags, integer implementation	8
Module 3: Dictionary-Based Data Compression Static dictionary, diagram coding, adaptive dictionary methods (LZ77, LZ78), applications in image compression.	8
Module 4: Advanced Lossless Compression Techniques Prediction by Partial Matching (PPM), Burrows-Wheeler Transform, Move-to-Front coding, JPEG-LS, CALIC.	8
Module 5: Lossy Data Compression and Multimedia Standards Distortion criteria, scalar quantization (uniform/adaptive), vector quantization, Linde-Buzo-Gray algorithm, overview of multimedia compression standards.	8

Text Books:

1. Khalid Sayood, Introduction to Data Compression, 5th Edition, Morgan Kaufmann (Elsevier), 2017.
2. David Salomon and Giovanni Motta, Data Compression: The Complete Reference, Springer, 4th ed. 2007.
3. Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, 2nd Edition, John Wiley & Sons, 2006.

Reference Books:

1. Mark Nelson & Jean-Loup Gailly, The Data Compression Book, 1995
2. David J. C. MacKay, Information Theory, Inference, and Learning Algorithms, 1st Edition, Cambridge University Press, 2003.
3. Fred Halsall, Multimedia Communications: Applications, Networks, Protocols and Standards, 1st Edition, Addison-Wesley, 2001

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Group Discussion on modern compression standards

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	0	3	2	1	3	1	1
CO2	2	2	0	2	2	2	2	1
CO3	2	1	2	3	3	3	2	2
CO4	3	0	2	2	2	2	1	1
CO5	2	3	2	1	2	2	1	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26527

Course Title: Digital Signal Processing

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr. Sumit Srivastava, Dr. Saikat Chakraborty, and Dr. Nandini Kumari

COURSE OBJECTIVES

This course enables the students to:

1.	Understand the fundamental concepts of Digital Signal Processing, discrete-time signals, systems, and signal representation for computational problem-solving.
2.	Apply transform techniques such as Z-Transform, Discrete Fourier Transform (DFT), and Fast Fourier Transform (FFT) to analyze and process digital signals efficiently.
3.	Design and evaluate digital filters for signal enhancement, noise reduction, and multimedia processing applications.
4.	Explore multirate Digital Signal Processing techniques and their applications in speech, image, audio, communication systems, Internet of Things (IoT), and intelligent computing.
5.	Develop software-based solutions for Digital Signal Processing using Python or MATLAB and apply DSP concepts to emerging technologies such as Artificial Intelligence, Machine Learning, Computer Vision and IoT.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamentals of Digital Signal Processing, discrete-time signals, systems, and Z-transform.
CO2	Apply DFT and FFT techniques to analyze digital signals, perform frequency-domain analysis, and solve computational signal-processing problems.
CO3	Design and compare IIR and FIR digital filters to improve signal quality in multimedia and communication applications.
CO4	Apply multirate signal processing techniques, including decimation, interpolation, and sampling rate conversion, to practical problems in audio, speech, Image Processing, and communication systems.
CO5	Develop and implement basic Digital Signal Processing algorithms using Python or MATLAB for multimedia and intelligent computing applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Fundamentals of Digital Signal Processing Introduction to Digital Signal Processing, Analog and Digital Signals, Sampling and Quantization, Discrete-Time Signals and Sequences, Basic Signal Operations, Linear Shift Invariant (LSI) Systems, Stability and Causality, Difference Equations, Introduction to Z-Transform and Realization of Digital Filters. Applications: Digital Audio Processing, Voice Assistants, Music Streaming and Digital Communication Systems.	8
Module 2: Frequency Domain Analysis Discrete Fourier Transform (DFT), Properties of DFT, Linear Convolution using DFT, Fast Fourier Transform (FFT), Radix-2 DIT FFT, Inverse FFT, Spectrum Analysis and Frequency Domain Representation. Applications: Audio compression, speech recognition, biomedical signal processing.	8
Module 3: Digital Filters Introduction to Digital Filters, Butterworth and Chebyshev Filters, Bilinear Transformation, Characteristics of FIR Filters, FIR Filter Design using Window Techniques (Rectangular, Hamming, Hanning), Comparison of FIR and IIR Filters. Applications: Noise reduction, image enhancement, Speech Enhancement, audio filtering.	8
Module 4: Multirate Digital Signal Processing Introduction to Multirate DSP, Down Sampling, Decimation, Up Sampling, Interpolation, Sampling Rate Conversion, Applications in Audio, Speech, Image Processing, Multimedia, Wireless Communication, and IoT.	8
Module 5: Computational DSP and Emerging Applications Introduction to Python Libraries (NumPy, SciPy, Matplotlib), Signal Visualization, FFT-based Signal Analysis, Basic Filter Implementation, DSP Applications in Artificial Intelligence, Machine Learning, Computer Vision, Smart Healthcare, Voice Assistants, Wearable Devices and Intelligent Systems.	8

Text Books:

1. Richard G. Lyons, Understanding Digital Signal Processing, 3rd Edition, Pearson Education, 2010.
2. Sanjit K. Mitra, Digital Signal Processing: A Computer-Based Approach, 4th Edition, McGraw-Hill, 2010.

Reference Books:

1. John G. Proakis and Dimitris G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, 4th Edition, Pearson Education, 2007.
2. Alan V. Oppenheim and Ronald W. Schafer, Discrete-Time Signal Processing, 3rd

Edition, Pearson Education, 2009.
3. Vinay K. Ingle and John G. Proakis, Digital Signal Processing Using MATLAB, 3rd Edition, Cengage Learning, 2012.
4. Nagoorkani A., Digital Signal Processing, 2nd Edition, McGraw-Hill Education, 2017.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment:

- Student Feedback on Course Outcome

Course Delivery Methods:

Code	**Course Delivery Method**
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/Teaching Aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internet
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading:

0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	0	0	0	0	1
CO2	3	3	2	2	0	0	0	1
CO3	2	3	3	2	1	0	0	1
CO4	2	3	2	3	2	0	0	2
CO5	2	2	3	3	2	1	1	3

Mapping Between Course Outcomes and Course Delivery Method:

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6
CO2	CD1, CD2, CD3, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD7
CO4	CD1, CD2, CD3, CD4, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26529

Course Title: Graph Theory

Pre-requisite(s): Mathematics for Computing, Data Structures

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teacher: Dr Srabani Mukhopadhyaya, Dr B.K. Sarakar

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts, terminologies, representation, and structural properties of graphs.
2. Develop analytical skills to study and solve problems related to paths, circuits, trees, connectivity, and planarity in graphs using mathematical reasoning.
3. Familiarize with graph-theoretic concepts such as graph colouring, matching, covering, and network flow that are fundamental for several problems in the domain of Computer Science.
4. Model and analyze real-world problems using graph-theoretic concepts and techniques.
5. Apply graph-theoretic methods and their applications in computer science and related disciplines by enhancing students' logical reasoning and problem-solving skills

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts, terminology, representations, and structural properties of graphs.
CO2	Analyze graph properties related to paths, circuits, trees, connectivity, and planarity using appropriate mathematical techniques and theorems.
CO3	Apply graph-theoretic concepts such as spanning trees, cut-sets, connectivity, graph colouring, matching, and covering to solve mathematical and computational problems.
CO4	Evaluate graph structures, colouring principles, matching techniques, and planarity criteria for network and optimization problems.
CO5	Model and solve practical problems arising in computer science, communication networks, scheduling, and optimization using graph-theoretic concepts.

SYLLABUS

<table>
<tr><th>MODULE</th><th>NO. OF LECTURE HOURS</th></tr>
<tr><td>Module 1: Fundamental Concepts of Directed Graphs

Introduction, Application of graphs, Basic terminologies, Graphs with special properties: Simple graphs, multi-graphs, directed and undirect graphs, Null graph, Complete graphs, Regular graphs, Bipartite graphs, Weighted graphs; Degree of a vertex, Handshaking theorem, Subgraphs, Induced subgraphs, Cliques, Graph Isomorphism.

Basic terminologies related to Directed Graphs

Matrix Representation of Graphs: Adjacency matrix, Incidence matrix, Weight matrix (Definition and simple properties).</td><td>8</td></tr>
<tr><td>Module 2: Introducing Path, Circuit, Connectivity

Walks, Paths and Circuits, Connected graphs and Connected Components, Eulerian Trails, Eulerian Circuits, Eulerian Graphs and their characterizations, Hamiltonian Cycles, Hamiltonian Graphs: Necessary condition, Sufficient conditions (Dirac's Theorem, Ore's Theorem).

Connectivity for Directed graphs, Directed path, Semipath, Semiwalk, Semicircuit, Strongly connected graphs, Weakly connected graphs.</td><td>8</td></tr>
<tr><td>Module 3: Trees, Network Connectivity, and Flow Networks

Trees: Properties and characterization of trees, Distance and Centers in a tree, 2-ary tree, m-ary tree, Rooted tree, Spanning tree: Branches and Chords, Rank and Nullity, Fundamental circuits; Spanning trees in a weighted graph.

Cut-Sets and Bridges: Cut-sets, Cut-vertex/Articulation point, Bridges, Properties of Cut-sets, Fundamental Cut-sets, Fundamental Circuits and Cut-sets.

Network Flow Concepts, Max Flow - Min Cut Theorem</td><td>8</td></tr>
<tr><td>Module 4: Graph Connectivity and Planarity

Connectivity: Edge Connectivity, Vertex Connectivity, k-connected graphs, Separable graphs. Simple results related to connectivity. Finding Biconnected Components and Articulation points of a given graph.

Planarity of Graphs: Drawing graphs in a plane, Planar Graphs, Planar embeddings, Dual graphs, Outer Planar Graphs, Euler's formula, Kuratowski's theorem.</td><td>8</td></tr>
<tr><td>Module 5: Graph Coloring and Matching Theory</td><td>8</td></tr>
</table>

Graph Colouring: Vertex Colouring, Proper Colouring, Chromatic Number, Chromatic number of specific graphs, Chromatic number-maximum degree, Independent set, Independence Number-Chromatic number, Clique number, Dominating set, Chromatic Polynomial, Five Colour and Four Colour Theorems. Covering and Matching: Maximal Matching, Matching number, Vertex cover, Edge cover.	

Text Books:

1. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall of India Pvt. Ltd., New Delhi, 2016.
2. Douglas B. West, Introduction to Graph Theory, Prentice-Hall of India Pvt. Ltd., New Delhi, 1995.

Reference Books:

1. F. Harary, Graph Theory, Addison-Wesley, 2001
2. K.R. Parthasarathi, Basic Graph Theory, Tata McGraw-Hill Publ. Co. Ltd., New Delhi.
3. L.R. Foulds, Graph Theory Applications, Narosa Publishing House, 2nd Edition, New Delhi, 1991.
4. D. S. Malik, M. K. Sen and S. Ghosh, Introduction to Graph Theory, Cengage Learning, 2014.
5. T.H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, McGraw-Hill Book Company, 2013.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based assignments
CD3	Laboratory experiments/ Teaching aids/Seminars
CD4	Mini Projects
CD5	Interactive lectures, Flipped classroom activities
CD6	Self-learning through NPTEL materials and internets
CD7	Case-studies via research papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	0	1	0	1
CO2	3	3	3	2	1	1	0	1
CO3	2	3	3	2	1	1	0	1
CO4	3	3	3	3	2	2	0	2
CO5	3	3	3	3	2	3	0	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD2, CD3
CO3	CD1, CD4, CD5
CO4	CD1, CD5, CD6, CD7
CO5	CD1, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26531

Course Title: Modeling and Solving Optimization Problems

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): Analysis of Algorithms, Python in Linux

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr. Ritesh Jha, Dr. Rathindra Nath Dutta, Dr. B. K. Sharma

COURSE OBJECTIVES

This course enables the students to:

1. Introduce the fundamental concepts of optimization and its mathematical modeling.
2. Develop skills for formulating real-world problems using Linear Programming, Integer Programming etc.
3. Provide understanding of various optimization techniques such as graphical methods, Simplex method, duality theory, branch and bound, and approximation algorithms.
4. Enable students to model and solve practical decision-making problems.
5. Familiarize students with advanced optimization concepts including constrained optimization, convex optimization, and modern optimization solvers.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the concepts of optimization and mathematical modeling of different optimization problem types along with their feasibility, and optimality conditions.
CO2	Formulate and solve Linear Programming problems.
CO3	Develop Integer Linear Programming models for real-world combinatorial problems.
CO4	Apply optimization techniques to solve complex optimization problems.
CO5	Analyze and solve common convex and nonlinear optimization problems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Optimization and Mathematical Modeling: Introduction to optimization problems and various real-world optimization problems; unconstrained and constrained optimization Problems, optimization variables, objective function, constraints; overview of feasible, infeasible, and optimal Solutions; Different Types of optimization problems (with examples): Linear Program (LP), Integer Linear Program (ILP), Binary Integer Program (BIP), Mixed-Integer Linear Program (MILP), Nonlinear Program, Quadratic Program (QP), Mixed-Integer Nonlinear Program (MINLP); Convex vs Nonconvex Function, Local vs Global Optima, Static vs Dynamic Optimization;	8
Module 2: Solving Linear Programming Problems: Standard and canonical forms, geometric interpretation of feasible Region; types of solutions: infeasible and feasible, bounded and unbounded, unique solution and infinitely many solutions; solving LP with graphical method (up to 2 variables); Simplex method; handling special cases in Simplex method; conversion between primal and dual formulations, weak and strong duality theorems; overview of computational complexity of Simplex Method and other polynomial-time methods.	8
Module 3: Formulating Integer Linear Programs Modeling common logical/programming constructs as linear constraints: mutual exclusivity between binary variables (either-or, selecting exactly one option), selecting at most/at least/exactly K-out-of-N binary choices, dependency between binary variables (logical-OR, logical-AND, implication/if condition, if and only if), general if-else conditions, multiplication of binary variables; Modeling Real Problems as ILP such as sorting problem, 0/1 knapsack problem, assignment problem, facility location problem, job scheduling, transportation problem, vertex cover, independent set, bipartite matching, bin-packing, set cover.	8
Module 4: Solving Integer Linear Programs LP relaxation, branch and bound method with examples, optimality gap; overview of computational hardness of ILP, modern solvers and techniques; LP rounding, vertex cover 2-approximation algorithm, set cover *f*-approximation	8
Module 5: Advanced Optimization Unconstrained optimization, necessary and sufficient conditions for optimality; Overview of constrained optimization, Lagrange multipliers, Karush-Kuhn-Tucker (KKT) conditions; convex optimization general form, LP and least-squares	8

problems as special cases of convex optimization; examples of non-linear optimizations such as machine learning; modern solvers and techniques;	

Text Books:

1. Dimitris Bertsimas and John N. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, 1997.
2. H. Paul Williams, Model Building in Mathematical Programming, 5th Edition, John Wiley & Sons.
3. Stephen Boyd and Lieven Vandenberghe, Convex Optimization, Cambridge University Press (https://web.stanford.edu/~boyd/cvxbook).

Reference Books:

1. Hamdy A. Taha, Operations Research: An Introduction, 11th Edition, Pearson, 2024.
2. David G. Luenberger and Yinyu Ye, Linear and Nonlinear Programming, Springer.
3. George L. Nemhauser and Laurence A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons.
4. Jiří Matoušek and Bernd Gärtner, Understanding and Using Linear Programming, Springer, 2006.
5. David B. Shmoys and David P. Williamson, The Design of Approximation Algorithms, Cambridge University Press (https://www.designofapproxalgs.com).

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Teaching aids/Seminars
CD4	Mini Projects
CD5	Expert talks
CD6	Self-learning, such as the use of NPTEL materials and the internet
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	0	0	0	2
CO2	3	3	3	3	0	1	0	2
CO3	3	3	3	3	2	1	0	2
CO4	3	3	3	3	2	2	1	3
CO5	3	3	3	3	0	0	0	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD4, CD7
CO2	CD1, CD2, CD3, CD4, CD6, CD7
CO3	CD2, CD3, CD4, CD6, CD7
CO4	CD2, CD3, CD4, CD7
CO5	CD1, CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26533

Course Title: Fuzzy Computing

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): None

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr. Jaya Pal, Dr. Shripal Vijayvargiya, Dr. Amrita Sarkar

COURSE OBJECTIVES

This course enables the students to:

1. Introduce students to the concept of fuzzy sets, membership functions, and how they differ from classical (crisp) sets and Boolean logic.
2. Enable students to perform fuzzy logic operations such as union, intersections, complements, and apply fuzzy relations and compositions in problem-solving.
3. Equip students with the ability to design Mamdani and Takagi–Sugeno–Kang (TSK) type fuzzy inference systems and apply them to real-world decision-making problems.
4. Make students proficient in converting crisp inputs to fuzzy values (fuzzification) and converting fuzzy outputs back to crisp values (defuzzification) using methods like centroid, mean of maxima, etc.
5. Train students to construct IF-THEN fuzzy rule bases and linguistic variable models for solving engineering and AI problems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Define and explain fuzzy sets, membership functions, fuzzy relations, and distinguish between crisp and fuzzy logic systems.
CO2	Perform and analyze fuzzy logic operations such as union, intersection, complement, Cartesian product, and composition of fuzzy relations to solve computational problems.

CO3	Design Mamdani and Takagi–Sugeno–Kang (TSK) based Fuzzy Inference Systems (FIS) using appropriate fuzzification, rule evaluation, aggregation, and defuzzification methods.
CO4	Build and evaluate fuzzy IF-THEN rule bases and linguistic variable models for knowledge representation and decision-making in uncertain environments.
CO5	Analyze and implement fuzzy computing techniques in practical applications such as fuzzy control systems, pattern recognition, medical diagnosis, image processing, and intelligent systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module I: Introduction to Fuzzy Logic, Classical Sets and Fuzzy sets Introduction to Fuzzy Logic, Classical Sets (Crisp Sets), Operation on Classical Sets- Union, Intersection, Complement, Difference; Properties of Classical Sets, Characteristic Function. Fuzzy Sets, Representation of fuzzy set, Fuzzy Sets Operations- Union, Intersection, Complement, More Operations on Fuzzy Sets, Properties of Fuzzy Sets, Fuzzy Numbers	6
Module II: Classical Relations and Fuzzy Relations Cartesian product, Crisp Relations-Cardinality of Crisp Relations, Operations on Crisp Relations, Properties of Crisp Relations Composition. Fuzzy relations: Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Cartesian Product and Composition of fuzzy relations (Max-Min Compositions and Max-Product compositions), Extension Principle, Fuzzy Arithmetic, Linguistic Variables and Hedges.	8
Module III: Membership Functions, Fuzzification and Defuzzification Features of Membership Functions- Core, Support, Boundary, Height, and α-cuts for Fuzzy Sets & Fuzzy relations, Strong α-cuts, Normal Fuzzy Set, and Convex Fuzzy Set. Various forms of Membership Functions, Fuzzification, Defuzzification- Definition, Defuzzification Methods- Centroid Method, Center of Sums, Mean of Maxima (MOM), Weighted Average.	8
Module IV: Fuzzy Logic and Approximate Reasoning Classical Logic: Propositional and Predicate Logic, Quantifiers, Classical inference rules, inference in Predicate Logic. Fuzzy logic: fuzzy propositions and their interpretations in terms of Fuzzy Sets, Fuzzy rules and their interpretations in terms of Fuzzy Relations, Fuzzy Inference or Approximate Reasoning.	9
Module V: Fuzzy Rule-based inference systems (FIS)	9

Fuzzy If-Then Rules-Rules Formulation, Mapping rules and Database/Rule Base composition, Architecture of Fuzzy Inference System, Architectural models-Mamdani Fuzzy Model, Takagi – Sugeno-Kang (TSK) Fuzzy Model, Design of Fuzzy Control Systems, Basic concepts of Type-II fuzzy set, Type I Fuzzy Set versus Type-II Fuzzy Set and Fuzzy C-means Clustering. Applications of Fuzzy Rule-based systems in intelligent decision support systems.	

Text Books:

1. Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, 4th Edition, Wiley Publications, 2021.
2. S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing*, 3rd Edition, Wiley India, 2018.

Reference Books:

1. S. Rajasekaran and G. A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications, 2nd Edition, Prentice Hall of India, 2017.
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, 1st Edition, Pearson, 2015.
3. J. S. R. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, 1st Edition, Pearson, 2015.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Chalk and Board / Traditional Lecture Method
CD2	Power Point Presentations (PPT)
CD3	E-Learning (NPTEL, Coursera, YouTube)
CD4	Problem Solving / Tutorial Sessions
CD5	Case Study Based Learning
CD6	Practical / Laboratory / Simulation Tools (MATLAB, Python)
CD7	Group Discussion / Seminar

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading:0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	1	0	0	2
CO2	3	3	2	2	1	0	0	2
CO3	2	3	3	3	2	1	0	2
CO4	2	3	3	2	2	1	1	2
CO5	2	2	3	3	3	2	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD3, CD4, CD6
CO4	CD1, CD2, CD4, CD5, CD7
CO5	CD2, CD3, CD5, CD6, CD7

TRACK: MODERN COMPUTING PARADIGM

COURSE INFORMATION SHEET

Course Code: CA26535

Course Title: Data Warehouse and Data Mining

Pre-requisite(s): Database Management System, Mathematics for Computing

Co-requisite(s): Python Programming

Credits: 03 L: 03 T:0 P:0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr Shashank Pushkar, Dr Aditi Panda, Dr Mansi Gupta

COURSE OBJECTIVES

This course enables the students to:

1.	Provide a strong foundation in the concepts, processes, functionalities, and applications of data mining, along with an understanding of different types of data and their characteristics
2.	Develop the ability to preprocess, transform, and reduce data using appropriate techniques to improve data quality and prepare datasets for mining.
3.	Introduce the concepts of data warehousing, multidimensional data models, OLAP operations, and data cube computation for efficient analytical processing.
4.	Equip students with the knowledge of association rule mining, frequent pattern mining techniques, and pattern evaluation methods for discovering useful knowledge from data.
5.	Enable students to apply classification, prediction, and clustering techniques, and evaluate their suitability for solving data mining problems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the concepts, functionalities, processes, and applications of data mining, and analyze different types of data using appropriate statistical and similarity measures.
CO2	Apply data preprocessing techniques, including data cleaning, integration, transformation, dimensionality reduction, and sampling, to prepare datasets for effective data mining.
CO3	Explain the architecture, multidimensional models, OLAP operations, and data cube concepts used in data warehousing for analytical data processing.

CO4	Apply association rule mining techniques and frequent pattern mining algorithms, and evaluate the significance of discovered patterns using appropriate pattern evaluation measures.
CO5	Apply supervised and unsupervised learning techniques, including k-nearest neighbor and k-means clustering, and evaluate their performance using appropriate classification metrics.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction and Understanding Data Introduction to Data Mining, Steps in Knowledge Discovery, Data Mining Functionality, Classification of Data Mining Systems, and Major Issues in Data Mining. Understanding Data: Types of data, Statistics of Data, Similarity and Dissimilarity Measures.	8
Module 2: Data Preprocessing Data Quality Measures, Data Cleaning, Data Integration, Data Transformation: Normalization, Discretization, Data Compression, Sampling; Dimensionality Reduction: Principal Component Analysis, Attribute Subset Selection.	8
Module 3: Data Warehouse and Data Cube Introduction, A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Modeling: Schema and Measures, Types of OLAP operations, Data Cube Computation.	8
Module 4: Pattern Mining and Evaluation Methods Basic Concepts: Association Rule and its types, Market Basket Analysis, Frequent Itemset, Closed Itemset, and Association Rules Frequent Itemset Mining Methods: Apriori Algorithm, Generating association rules from frequent itemsets, Improving the efficiency of Apriori, Pattern-growth approach for mining frequent itemsets Pattern evaluation methods: Which patterns are interesting, From association analysis to correlation analysis, A comparison of pattern evaluation measures	8
Module 5: Classification and Prediction Classification & Prediction general concept and methods, Types of learning, Supervised Learning: k-nearest-neighbor classifiers; Metrics for evaluating	8

classifier performance; Unsupervised Learning: What is cluster analysis? Overview of basic clustering methods, k-Means: a centroid-based technique.	

Text books:

1. Jiawei Han & Jian Pei “Data Mining Concepts & Techniques”, Publisher Harcourt India. Private Limited, 4th Edition, 2022.

Reference books:

1. Gupta G.K. Introduction to Data Mining with case Studies, PHI, New Delhi, 2006.
2. Berson A. & Smith S.J. Data Warehousing Data Mining, COLAP, TMH, New Delhi, 2004.
3. Dunham H.M. & Sridhar S. Data Mining, Pearson Education, New Delhi, 2006.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Tutorial/Assignments
CD3	Seminars/ Quiz

CD4	Mini Projects/ Projects
CD5	Laboratory Experiments/Teaching Aids
CD6	Online materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	3	2	3	3	1	1
CO2	3	2	2	2	2	1	1	1
CO3	3	2	2	3	3	3	2	1
CO4	3	2	2	1	1	2	1	1
CO5	3	2	3	2	3	2	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD3, CD5
CO2	CD1, CD3, CD5
CO3	CD1, CD3, CD5
CO4	CD1, CD3, CD5, CD4
CO5	CD1, CD3, CD5, CD4

COURSE INFORMATION SHEET

Course Code: CA26537

Course Title: Information Retrieval

Pre-requisite(s): Data Structures

Co- requisite(s): Python Programming

Credits: 03 L: 03 T: 0 P: 0

Class schedule per week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of Teacher: Dr Shah Fahad, Dr Richa

COURSE OBJECTIVES

This course enables the students to:

1.	Understand the fundamental concepts, architecture, and components of information retrieval systems and search engines.
2.	Learn how data is pre-processed and indexed for fast and accurate information retrieval.
3.	Apply retrieval models, term weighting techniques, and ranking algorithms to retrieve relevant information efficiently.
4.	Explore techniques for measuring the effectiveness and relevance of search engine results.
5.	Provide a foundational understanding of language modelling approaches in information retrieval.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand the basic concepts of information retrieval
CO2	Apply text preprocessing and indexing techniques to organize documents for efficient information retrieval.
CO3	Analyze and apply retrieval models and ranking techniques in information retrieval systems.
CO4	Perform the evaluation and critical analysis of the performance of the retrieval system
CO5	Describe the principles of language modelling in Information Retrieval

SYLLABUS

MODULE	(NO. OF LECTURE HOURS)
Module 1: Introduction Introduction; Search Engine Architecture; An overview of crawling, text transformation, index creation, user interaction, ranking, link analysis, evaluation, and deep web.	6
Module 2: Pre-processing and Indexing Pre-processing: tokenization, stop words, normalization, stemming, wildcard queries, spelling correction – edit distance and k-gram; Indexing: Index construction; Index compression.	10
Module 3: Scoring Parametric and zone indexes; term frequency and weighting; vector space model; efficient scoring and ranking; page ranking algorithms.	8
Module 4: IR Evaluation and Query Expansion Evaluation; Standard test collection; Evaluation of unranked and ranked retrieval; Assessing relevance; System quality and user utility, Query Expansion and Relevance Feedback.	8
Module 5: Language Model for Information Retrieval Language Models; The query likelihood model; Language modelling versus other approaches in IR; Extended language modelling approaches.	8

Text book:

1. Manning, Christopher D., Raghavan Prabhakar, and Schütze Hinrich, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2008.

Reference books:

1. Grossman David A., Frieder Ophir, Information Retrieval: Algorithms and Heuristics, Springer, 2004.
2. Croft Bruce, Metzler Donald, and Strohman Trevor, Search Engines: Information Retrieval in Practice, Pearson Education, 2009.
3. Ricardo Baeza-Yates and Neto Berthier Ribeiro, Modern Information Retrieval, 2nd Edition, Addison-Wesley, 2011.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

COURSE DELIVERY METHODS

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP
CD2	Projectors
CD3	Tutorials/Assignments
CD4	Seminars
CD5	Mini projects
CD6	Expert talks
CD7	Self- learning such as use of NPTEL materials and

Mapping of Course Outcomes onto Program Outcomes and Program-Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	2	0	1	3	3
CO2	3	3	3	3	1	2	0	2
CO3	3	3	3	3	2	3	2	3
CO4	3	3	3	2	2	3	1	3

CO5	3	1	2	2	1	1	2	2

MAPPING BETWEEN COURSE OUTCOMES AND COURSE DELIVERY METHOD

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD3
CO2	CD1, CD2, CD3
CO3	CD1, CD2, CD3
CO4	CD1, CD2, CD3
CO5	CD1, CD2, CD3

<u>**COURSE INFORMATION SHEET**</u>

Course Code: CA26539

Course Title: Data Visualization using Python

Pre-requisite(s): Data Structures

Co-requisite(s): Python Programming

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr Shruti Garg, Dr. Akriti Nigam, Dr. Mohua Banerjee

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts, principles, and importance of data visualization in data analytics and decision-making.
2. Develop knowledge of data preparation, exploratory data analysis, and visualization techniques using Python libraries.
3. Analyze and select appropriate visualization methods for different types of data and analytical problems.
4. Design effective dashboards and visualizations for communicating insights using modern visualization frameworks.
5. Evaluate visualization practices and apply data storytelling techniques to real-world applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts, principles, and significance of data visualization and exploratory data analysis.
CO2	Apply data preparation and preprocessing techniques to organize datasets for effective visualization.
CO3	Analyze and interpret statistical and interactive visualizations using Python-based visualization libraries such as Matplotlib, Seaborn, and Plotly.

CO4	Design appropriate visualization solutions and dashboards by applying visualization principles and best practices.
CO5	Evaluate visualization techniques and communicate analytical insights effectively through data storytelling for real-world applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Data Visualization Fundamentals Introduction to Data Visualization and Visual Analytics, Types of Data and Data Representation, Principles of Effective Data Visualization, Python Environment for Data Visualization (Anaconda, Jupyter Notebook), Introduction to NumPy and Pandas, Data Import from CSV, Excel, JSON, and SQL, Data Cleaning and Preparation Basics.	8
Module 2: Exploratory Data Analysis and Data Visualization Exploratory Data Analysis (EDA), Descriptive Statistics and Data Summarization, Fundamentals of Matplotlib, Line, Bar, Pie, and Histogram Charts, Scatter and Box Plots, Plot Customization (Titles, Labels, Legends, Annotations), Multiple Plots and Figure Layouts.	8
Module 3: Statistical Data Visualization with Seaborn Introduction to Seaborn, Distribution and Density Plots, Categorical Data Visualization, Pair Plots and Regression Plots, Correlation Analysis and Heatmaps, Themes, Color Palettes, and Figure Styling, Comparative Visualization Techniques.	8
Module 4: Interactive Visual Analytics Introduction to Plotly, Interactive Charts using Plotly Express, Dashboard Design Principles, Dashboard Development using Dash, Geospatial Visualization, Time-Series Visualization, Introduction to Streamlit for Dashboard Deployment.	8
Module 5: Visual Analytics and Data Storytelling Principles of Data Storytelling, Visualization Ethics and Best Practices, Dashboard Evaluation and Interpretation, Business Intelligence and Decision Support, Visualization in Healthcare, Finance, and Social Media, Case Studies in Data Visualization, Emerging Trends in AI-Assisted Visualization	8

Text Books

1. McKinney, W., Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media, 2022.

2. VanderPlas, J., Python data science handbook: Essential tools for working with data (2nd ed.). O'Reilly Media, 2022.
3. Grus, J., Data science from scratch: First principles with Python (2nd ed.). O'Reilly Media, 2019.

Reference Books

1. Wilke, C. O., Fundamentals of data visualization: A primer on making informative and compelling figures. O'Reilly Media, 2019.
2. Kirk, A., Data visualization: A handbook for data-driven design (2nd ed.). SAGE Publications, 2019.
3. Munzner, T., Visualization analysis and design. CRC Press, 2014.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Classroom lectures using board and LCD projector presentations
CD2	Tutorials, assignments, and problem-solving sessions
CD3	Case studies, group discussions, and seminar presentations
CD4	Demonstration of Python-based data visualization tools and dashboards
CD5	Mini-projects, case analysis, and data visualization design exercises
CD6	E-learning resources, MOOCs, and online documentation (Python, Matplotlib, Seaborn, Plotly)
CD7	Video lectures, research paper discussions, and self-learning activities

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	1	0	0	1
CO2	3	3	2	2	2	1	0	1
CO3	3	3	3	3	2	1	0	1
CO4	3	3	3	3	3	2	1	2
CO5	3	3	2	3	2	2	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6
CO2	CD1, CD2, CD4, CD6
CO3	CD1, CD3, CD4, CD6
CO4	CD1, CD3, CD4, CD5, CD6
CO5	CD1, CD3, CD5, CD6, CD7

TRACK: Web Application

COURSE INFORMATION SHEET

Course Code: CA26541

Course Title: Introduction to NoSQL Backends

Pre-requisite(s): Database Management System, Data Structures

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr. Debjani Mustafi, Dr. Sathish Chander

COURSE OBJECTIVES

This course enables the students to:

1.	Introduce the fundamental concepts, architectures, and data models of NoSQL databases.
2.	Understand distributed storage systems, replication, and consistency mechanisms in NoSQL databases.
3.	Learn Map-Reduce programming and key-value based data processing techniques.
4.	Familiarize with document databased and column-family databases along with their applications.
5.	Understand graph databases and schema evolution in NoSQL systems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the concepts, data models and design principles of NoSQL databases for scalable data management.
CO2	Analyze distributed NoSQL architectures, replication strategies and consistency models for reliable data storage.
CO3	Apply Map-Reduce programming and key-value database concepts to process and manage large-scale data efficiently.
CO4	Evaluate document-oriented and column-family databases for suitable application domains and use cases.

CO5	Assess graph databases and schema evolution techniques to support complex relationships and changing data requirements.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of NoSQL Databases Introduction to NoSQL, Impedance Mismatch, Aggregate Data Models, Aggregates, Key-Value and Document Data Models, Column-Family Stores, Relationships, Graph Databases, Schemaless Databases, Materialized View, Modelling for Data Access.	10
Module 2: Distributed NoSQL Architecture and Consistency Distribution models, Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication, Update Consistency, Read Consistency, Relaxing Consistency, Relaxing Durability, Quorums.	10
Module 3: Distributed Data Processing and Key-Value Stores Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, Key-Value Databases, Key-Value Store, Store Features, Suitable Use Cases.	6
Module 4: Document and Column-Family Databases Document Databases, Features, Suitable Use Cases, Colum-Family Stores, Features, Suitable Use Cases.	6
Module 5: Graph Databases and Schema Evolution Graph Databases, Features, Suitable Use Cases, Schema Migration, Schema Changes, Schema Changes in RDBMS, Schema Changes in NoSQL Data Store.	8

Text Books:

1. Pramod J. Sadalage, Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st Edition, Addison-Wesley Professional, 2012, ISBN: 978-0321826626.

Reference Books:

1. Brian Pau, NoSQL Databases: Comprehensive Introduction to Non-Relational Data Management, 1st Edition, Khin Soe, 2024.

2. Michael Kaufmann, Andreas Meier, SQL and NoSQL Databases: Modeling, Languages, Security and Architectures for Big Data Management, 2nd Edition, Springer Nature, 2023.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	1	1	0	1
CO2	3	2	3	3	1	0	0	1

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO3	3	2	3	3	1	0	0	1
CO4	2	2	3	3	1	0	2	2
CO5	3	3	3	3	1	0	0	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD5, CD6, CD7
CO2	CD1, CD2, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6
CO5	CD1, CD2, CD3, CD4, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26543

Course Title: Frontend Design 1

Pre-requisite(s): Web Application Development Skills using HTML and CSS

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr Sandip Ghosal, Prof. Abhijit Mustafi

COURSE OBJECTIVES

This course enables the students to:

1.	Develop understanding of CSS concepts for designing modern web interfaces.
2.	Use advanced CSS selectors, layouts, animations, and design techniques.
3.	Learn responsive web design principles using Flexbox, Grid, and media queries.
4.	Develop skills to create reusable UI components and maintainable style architectures.
5.	Apply CSS frameworks, preprocessors, accessibility, and performance optimization techniques.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand advanced CSS concepts, selectors, and styling mechanisms.
CO2	Design responsive web layouts using Flexbox, Grid, and media queries.
CO3	Develop user interfaces using CSS transitions, animations, and transformations.
CO4	Apply modern CSS architecture, frameworks, and UI design practices.
CO5	Optimize web interfaces considering accessibility, performance.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Advanced CSS Fundamentals CSS syntax and cascade, specificity, inheritance, advanced selectors, pseudo classes and pseudo elements, CSS variables, units and measurements, box model, positioning, overflow handling.	8
Module 2: Modern CSS Layout Techniques Responsive web design principles, Flexbox layout model, Grid layout system, alignment and spacing, multi-column layouts, media queries, mobile-first design approach, viewport-based design.	8
Module 3: CSS Effects and Interactive Design CSS transitions, transformations, animations, keyframes, filters, shadows, gradients, clipping and masking, creating interactive components using advanced CSS techniques.	8
Module 4: CSS Architecture and Frameworks CSS organization methodologies, reusable components, CSS preprocessors introduction, utility-first CSS concepts, introduction to Bootstrap/Tailwind CSS, customization and theming.	8
Module 5: Advanced UI Design Practices Accessibility in CSS, cross-browser compatibility, CSS performance optimization, responsive images, design systems, best practices for maintainable frontend development.	8

Text Books:

1. Jon Duckett, HTML and CSS: Design and Build Websites, Wiley, 2025.
2. Eric A. Meyer and Estelle Weyl, CSS: The Definitive Guide, O'Reilly Media, 2025
3. Keith J. Grant, CSS in Depth, Manning Publications, 2023

Reference Books:

1. Lea Verou, CSS Secrets, O'Reilly Media, 2015.
2. Ben Frain, Responsive Web Design with HTML5 and CSS, Packt Publishing, 2022.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Classroom Lectures
CD2	Interactive Discussions and Question–Answer Sessions
CD3	Demonstration of CSS Concepts and Case Studies
CD4	Laboratory Demonstrations and Hands-on Practice
CD5	Tutorials and Problem-Solving Sessions
CD6	Assignments, Quizzes, and Self-Learning Activities
CD7	Seminar Presentations and Project-Based Learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	0	0	1	2
CO2	3	3	3	3	1	0	1	2

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO3	2	3	3	3	2	0	1	2
CO4	2	2	3	3	2	1	2	3
CO5	2	2	2	3	2	1	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD3, CD4, CD5
CO3	CD1, CD3, CD4, CD5, CD6
CO4	CD1, CD3, CD4, CD6
CO5	CD1, CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26545

Course Title: Introduction to Web Scraping

Pre-requisite(s): Web application Development using HTML, CSS

Co-requisite(s): Python in Linux

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: II / V

Branch: MCA

Name of the Teachers: Dr. Jamimamul Bakas and Prof. Abhijit Mustafi

COURSE OBJECTIVES

This course enables the students to:

1. Design and implement a complete web scraping application to solve a real-world data collection problem.
2. Learn HTML parsing techniques using Python and BeautifulSoup.
3. Develop web crawlers to extract and organize data from websites.
4. Apply advanced web scraping techniques for handling forms, APIs, sessions, and responsible scraping.
5. Design and implement a real-world web scraping application.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamentals of web technologies, HTTP, HTML, and web scraping principles.
CO2	Extract structured information from web pages using Python Requests and BeautifulSoup.
CO3	Develop web crawlers to collect, process, and store data from multiple web pages.
CO4	Apply advanced web scraping techniques involving forms, sessions, APIs, and responsible scraping practices.
CO5	Design and implement a complete web scraping application to solve a real-world data collection problem.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Web Scraping- Introduction to Web Scraping and its applications, Web technologies: HTTP, URLs, HTML structure, CSS selectors, Python essentials for web scraping, Sending HTTP requests using requests, Inspecting web pages using browser Developer Tools.	8
Module 2: HTML Parsing using BeautifulSoup - Introduction to BeautifulSoup with installation guide, Parsing HTML documents, Finding elements using tags, IDs and classes, HTML tree navigation, Extracting text, links, images and tables, Handling missing elements and exceptions, Using Regular Expressions for element selection.	8
Module 3: Web Crawling and Data Storage - Introduction to Web Crawling, Single-page and multi-page scraping, Following hyperlinks automatically, Crawling multiple webpages, Exporting scraped data to CSV, Introduction to JSON data format, Basic data cleaning and preprocessing.	8
Module 4: Advanced Web Scraping Techniques - Crawling through forms, Handling cookies and sessions, HTTP authentication, Crawling through APIs-Introduction to Web APIs Parsing, JSON responses, Undocumented APIs, Combining APIs with Other Data Sources.	8
Module 5: Responsible and Ethical Web Scraping- Ethical considerations in web scraping, Responsible web scraping practices, robots.txt and Terms of Service, Avoiding common scraping traps- rate limiting, request headers, Common Form Security Features. Case Study: Design and implementation of a web scraping application.	8

Text Books:

1. Mitchell, Ryan. Web Scraping with Python: Collecting More Data from the Modern Web. 2nd ed., O'Reilly Media, 2018.

Reference Books:

1. Munzert, Simon, Christian Rubba, Peter Meißner, and Dominic Nyhuis. Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons, 2015.
2. Vanden Broucke, Seppe, and Bart Baesens. Practical Web Scraping for Data Science: Best Practices and Examples with Python. Apress, 2018.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Board / Slides
CD2	Tutorial / Assignments
CD3	Self Learning through NPTEL / online materials
CD4	Coding demonstration in class
CD5	Mini projects
CD6	Interactive Lectures, Flipped Classroom Activities
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	1	0	1	1	2
CO2	3	1	1	1	0	1	1	2

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO3	2	2	1	2	0	1	1	3
CO4	2	3	2	3	1	1	1	3
CO5	2	2	3	3	1	1	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD4
CO4	CD1, CD2, CD3, CD4, CD5
CO5	CD1, CD2, CD3, CD4, CD5

PROGRAM ELECTIVE II

TRACK: CORE COURSES

COURSE INFORMATION SHEET

Course Code: CA26605

Course Title: Image Processing

Pre-requisite(s): Digital Signal Processing, Mathematics for Computing

Co-requisite(s): None

Credits: 03 L:03 T:0 P:0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teacher: Dr Akriti Nigam, Dr. Ravi Shankar Mehta

COURSE OBJECTIVES

This course enables the students to:

1. Provide a strong foundation in digital image processing concepts, image representation, and the fundamental operations involved in digital image analysis.
2. Develop the ability to enhance digital images using spatial and frequency domain techniques for improving image quality.
3. Introduce image restoration techniques and noise models for recovering degraded images using appropriate filtering methods.
4. Equip students with the knowledge of image segmentation techniques for extracting meaningful information from digital images.
5. Enable students to understand and apply image compression techniques for efficient storage and transmission of digital images.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts of digital image processing, image representation, sampling, quantization, and pixel relationships.
CO2	Apply spatial and frequency domain techniques, including histogram processing and filtering methods, to enhance digital images.
CO3	Analyze image degradation models and apply appropriate restoration techniques to reduce noise and recover degraded images.

CO4	Apply edge detection and image segmentation techniques to identify and extract meaningful regions and objects from digital images.
CO5	Explain the principles of image compression and apply lossless and lossy compression techniques, including JPEG, for efficient image storage and transmission.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Image Representation and Processing Fundamentals Introduction to Digital Image Processing (DIP): definitions, applications, and fundamental steps. Components of an image processing system Digital image representation; sampling and quantization. Basic relationships between pixels: neighbors, connectivity, distance measures.	8
Module 2: Image Enhancement Techniques Spatial domain methods: basic gray-level transformations. Histogram processing: equalization and specification (matching). Spatial filtering fundamentals: smoothing (mean, median) filters. Sharpening spatial filters: Laplacian and gradient-based operators. Review of the Fourier Transform; the 2-D Discrete Fourier Transform (DFT). Filtering in the frequency domain: ideal, Butterworth, and Gaussian low-pass/high-pass filters. Homomorphic filtering.	10
Module 3: Image Restoration and Noise Removal Image degradation model and noise models (Gaussian, salt-and-pepper, Rayleigh). Restoration using spatial filters (mean and order-statistics filters). Periodic noise reduction; inverse filtering and Wiener filtering.	8
Module 4: Edge Detection and Image Segmentation Edge detection: gradient operators (Sobel, Prewitt) and the Canny edge detector. Thresholding techniques: global and adaptive thresholding. Region-based segmentation: region growing, splitting and merging. Watershed segmentation algorithm.	8
Module 5: Image Compression Techniques Fundamentals of image compression; coding, interpixel, and psychovisual redundancy. Lossless compression: Huffman coding, run-length coding, LZW coding. Lossy compression: transform coding fundamentals, Discrete Cosine Transform (DCT). The JPEG compression standard.	8

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018.
2. S. Jayaraman, S. Esakkirajan, and T. Veerakumar, Digital Image Processing, Tata McGraw Hill, 2009.

Reference Books:

1. Milan Sonka, Vaclav Hlavac, and Roger Boyle, Image Processing, Analysis, and Machine Vision, 4th Edition, Cengage Learning, 2014.
2. William K. Pratt, Digital Image Processing, 4th Edition, Wiley-Interscience, 2007.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities

Code	Course Delivery Method
CD6	Self- learning through NPTEL materials and internet
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	1	2	1	0	1
CO2	3	3	3	2	2	2	1	1
CO3	3	3	3	3	2	2	2	1
CO4	3	3	3	3	2	3	2	2
CO5	3	3	2	2	3	3	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6
CO2	CD1, CD2, CD3, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26607

Course Title: Cloud Computing

Pre-requisite(s): Data Communication and Computer Network, Operating Systems

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr Kuntal Mukherjee, Dr Sanchita Paul

COURSE OBJECTIVES

This course enables the students to:

1. Understand cloud computing concepts, architectures, service models and deployment models.
2. Analyze virtualization, containerization and cloud infrastructure technologies.
3. Design and deploy scalable cloud-based solutions and services.
4. Apply cloud security, governance, compliance and SLA mechanisms.
5. Evaluate emerging cloud technologies, multi-cloud systems and enterprise applications

COURSE OUTCOMES (COs)]

After the completion of this course, students will be able to:

CO1	Explain cloud computing concepts, characteristics, service and deployment models.
CO2	Apply virtualization and container technologies for cloud resource management.
CO3	Design scalable cloud architectures and services.
CO4	Evaluate cloud security, privacy, governance and SLA mechanisms.
CO5	Analyze modern cloud applications and emerging cloud paradigms.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Cloud Computing Evolution of Cloud Computing; NIST Definition and Essential Characteristics; Cloud Service Models (IaaS, PaaS, SaaS, FaaS); Cloud Deployment Models (Public, Private, Hybrid, and Community Clouds); Cloud Economics; Comparative Overview of Major Cloud Platforms (AWS, Microsoft Azure, and Google Cloud Platform).	8
Module 2: Virtualization and Cloud Infrastructure Virtualization Concepts and Architecture; Hypervisors and Virtual Machine Management; Virtual Machine Provisioning and Migration; Storage and Network Virtualization; Containerization using Docker; Container Orchestration using Kubernetes; Scalability, Elasticity, and Resource Management in Cloud Environments.	9
Module 3: Cloud Architecture and Platforms Service-Oriented Architecture (SOA); RESTful Web Services and APIs; Microservices Architecture; Cloud Storage Architectures; Distributed Data Processing using Hadoop and MapReduce; Cloud Workflow Management Systems; Cloud Application Migration Strategies.	7
Module 4: Cloud Security, Governance, and Service Level Agreements (SLAs) Identity and Access Management (IAM); Authentication and Authorization; Data Encryption and Privacy; Cloud Security Governance and Regulatory Compliance; Service Level Agreements (SLAs) and SLA Lifecycle Management; Business Continuity and Disaster Recovery.	8
Module 5: Cloud Applications and Emerging Trends Cloud Monitoring and Resource Management; Cloud Metering and Billing; Content Delivery Networks (CDNs); Edge and Fog Computing; Serverless Computing; Artificial Intelligence and Machine Learning as Cloud Services (AI/ML); Federated, Hybrid, and Multi-Cloud Architectures.	6

Text Books:

1. Erl. Thomas, Eric Barceló Monroy. Cloud Computing: Concepts, Technology, Security, and Architecture. 2nd ed. Pearson, 2023. ISBN-13: 978-0138052188

2. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski (Eds.), Cloud Computing: Principles and Paradigms. Wiley, 2011. ISBN-13: 978-0470887998;

Reference Books:

1. Barrie Sosinsky, Cloud Computing Bible. Wiley Publishing, Indianapolis, Indiana, USA, 2011. ISBN-13: 978-0470903568; ISBN-10: 0470903562.

2. Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Morgan Kaufmann Publishers (Elsevier), Waltham, MA, USA, 2012. ISBN-13: 978-0123858801

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Classroom Lectures
CD2	Smart Board/PPT
CD3	Case Studies
CD4	Cloud Demonstrations
CD5	Seminars

Code	Course Delivery Method
CD6	Assignments
CD7	Mini Projects

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	2	1	1	1
CO2	2	3	2	1	3	1	1	1
CO3	2	2	3	2	3	1	1	1
CO4	2	3	2	3	2	2	2	1
CO5	2	2	2	2	3	2	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD2, CD4
CO3	CD1, CD3, CD4, CD7
CO4	CD1, CD5, CD6
CO5	CD2, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26609

Course Title: Network Analysis

Pre-requisite(s): Graph Theory

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Abhijit Mustafi, Dr B.K. Sarkar, Dr Nandan Banerjee

COURSE OBJECTIVES

This course enables the students to:

1.	Understand the fundamental concepts, mathematical foundations, and topological properties of complex networks and graph theory.
2.	Analyze the structural characteristics of regular, random, small-world, and scale-free networks using quantitative measures.
3.	Apply network science principles and computational techniques to model and investigate real-world systems.
4.	Develop the ability to evaluate and compare different network models for solving practical problems.
5.	Foster analytical thinking and encourage the use of modern computational tools for network analysis and lifelong learning.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts of network science, graph theory, and various graph representations.
CO2	Analyze the structural characteristics and performance measures of regular and random networks.
CO3	Evaluate the properties, evolution, and applications of small-world and scale-free network models.
CO4	Apply appropriate computational techniques and modern tools to model and analyze complex networks.
CO5	Compare different network models and recommend suitable approaches for analyzing real-world complex systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS (42 Hours)
Module 1: Foundations of Network Science What is network science? General Principles. Set theoretic definition of a graph. Matrix algebra definition of a graph. The Konigsberg Graph. Spectral properties of graphs. Types of graphs. Topological structures.	8
Module 2: Network Structures and Characteristics Defining regular networks, Network Characteristics - Diameter, Centrality, and Average Path Length, Binary Tree Network, Toroidal Network, Hypercube Networks.	8
Module 3: Random Networks and Analysis Generation of Random Networks, Degree Distribution of Random Networks, Entropy of Random Networks, Properties of Random Networks, Weak Ties in Random Networks, Randomization of Regular Networks, Analysis.	8
Module 4: Small-World Network Models and Analysis Generating a Small-World Network, Properties of Small-World Networks, Phase Transition, Navigating Small Worlds, Weak Ties in Small-World Networks, Analysis.	8
Module 5: Analysis of Scale-Free Networks Generating a Scale-Free Network, Properties of Scale-Free Networks, Navigation in Scale-Free Networks, Analysis.	8

Text Books:

1. Lewis, T. G., Network Science: Theory & Practice, 1st Edition, John Wiley & Sons. Inc. Publications, 2009

Reference Books:

1. Newman M., Networks: An Introduction, 1st Edition, Oxford University Press, 2010

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of board/LCD projectors/OHPs
CD2	Concept-based Assignments
CD3	Laboratory Experiments/ Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures/Flipped Classroom Activities
CD6	Self-Learning through NPTEL materials and Internet resources
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	0	1	0	0	0	1
CO2	2	3	2	2	0	0	0	1
CO3	2	3	3	2	1	0	0	1
CO4	2	2	3	3	1	1	0	2
CO5	2	3	3	2	1	1	0	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD4, CD5, CD6
CO5	CD2, CD3, CD4, CD6, CD7

TRACK: MODERN COMPUTING PARADIGM

COURSE INFORMATION SHEET

Course Code: CA26611

Course Title: Evolutionary Computing

Pre-requisite(s): Mathematics for Computing, Analysis of Algorithms

Co-requisite(s): Machine Learning Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr Anup Kr. Keshri, Dr Ritesh Jha

COURSE OBJECTIVES

This course enables the students to:

1. Impart the knowledge in optimization.
2. Explore multi objective optimization.
3. Learn evolutionary algorithms.
4. Understand Evolutionary strategies and programming.
5. Get familiarized with the Multi-Objective Evolutionary algorithm.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Demonstrate principles of the optimization process.
CO2	Make use of multi-response optimization.
CO3	Apply the evolutionary programming.
CO4	Evaluate the process parameters for optimization.
CO5	Apply optimization techniques in mechanical component design.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Evolutionary Computing Introduction to optimization – single and multi-objective optimization – Evolutionary algorithms – principles of multi objective optimization.	8
Module 2: Convex Optimization Methods Convex programming, Karush-Kuhn-Tucker conditions, Direct functional evaluation, and derivative-based optimization techniques	8
Module 3: Evolutionary Computation and Metaheuristics Simulated annealing, Tabu search; NFL theorem; Biological principles of evolution, General scheme of EAs, Representation, Selection schemes, Population evaluation, Variation operators; Constraint handling; Schema theorem; Binary coded genetic algorithm, Real coded genetic algorithm.	8
Module 4: Evolutionary and Swarm Optimization Evolutionary strategies, Evolutionary programming, genetic programming, Differential evolution, Particle swarm optimization.	8
Module 5: Multi-Objective Evolutionary Algorithms Pareto-optimality, multi-objective evolutionary algorithms; Statistical analysis of EC techniques; Customization in EAs; Applications of multi-objective evolutionary algorithms – Mechanical component design – Truss-structure design – Other applications.	8

Text Books:

1. Back, T., Fogal, D. B. and Michalewicz, Z., Handbook of Evolutionary Computation, Oxford University Press, 1997.
2. Clerc, M.,Particle Swarm Optimization, ISTE, 2006.
3. Deb, K., Multi-objective Optimization using Evolutionary Algorithms, Wiley, 2001.

Reference Books:

1. Fogel, D. B., Evolutionary Computation, The Fossil Record, IEEE Press, 2003.
2. Goldberg, D., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.
3. Price, K., Storn, R. M., and Lampinen, J. A., Differential Evolution: A Practical Approach to Global Optimization, Springer, 2005.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Teaching aids/Seminars
CD4	Mini Projects
CD5	Expert talks
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	0	1
CO2	2	3	3	2	0	0	0	1
CO3	2	2	3	3	0	0	0	2
CO4	2	3	3	2	1	1	0	1
CO5	2	3	3	2	1	1	0	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD4, CD7
CO2	CD1, CD2, CD3, CD4, CD6, CD7
CO3	CD2, CD3, CD4, CD6, CD7
CO4	CD2, CD3, CD4, CD7
CO5	CD1, CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26613

Course Title: Reinforcement Learning

Pre-requisite(s): Mathematics for computing, Python programming

Co-requisite(s): Machine Learning Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr Nandan Banerjee, Dr Chandrashekhar Lavania

COURSE OBJECTIVES

This course enables the students to:

1. Introduce the mathematical foundations of sequential decision-making and reinforcement learning.
2. Develop an understanding of Markov Decision Processes and dynamic programming techniques.
3. Enable students to design and implement model-free and model-based reinforcement learning algorithms.
4. Study function approximation and deep reinforcement learning methods for large-scale problems.
5. Apply reinforcement learning techniques to solve real-world engineering and optimization problems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts and mathematical foundations of reinforcement learning and sequential decision-making.
CO2	Apply dynamic programming and Markov Decision Processes to compute optimal policies and value functions.
CO3	Analyze and implement model-free reinforcement learning algorithms for solving decision-making problems.
CO4	Evaluate policy optimization and deep reinforcement learning techniques for complex environments.

CO5	Design and develop intelligent agents using reinforcement learning techniques for real-world applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Reinforcement Learning Learning Paradigms (Supervised, unsupervised, reinforcement Learning), Elements of Reinforcement Learning, Agent, Environment, State, Action and Reward, Return and Discount Factor, Sequential Decision Making, Exploration versus Exploitation, Multi-Armed Bandit Problems, ε-Greedy, UCB and Softmax Policies, Applications of Reinforcement Learning	8
Module 2: Markov Decision Processes and Dynamic Programming Markov Property, Markov Reward Processes (MRP), Markov Decision Processes (MDP), State Value and Action Value Functions, Bellman Expectation and Bellman Optimality Equations, Policy Evaluation, Policy Improvement, Policy Iteration, Value Iteration, Generalized Policy Iteration (GPI), Curse of Dimensionality.	8
Module 3: Model-Free Reinforcement Learning Monte Carlo Methods, Temporal Difference Learning, TD(0) and TD(λ) , On-Policy and Off-Policy Learning, SARSA Algorithm, Q-Learning Algorithm, Expected SARSA, Eligibility Traces, Convergence Properties and Performance Analysis, Comparison of Model-Free Methods.	8
Module 4: Function Approximation and Deep Reinforcement Learning Motivation for Function Approximation, Linear Value Function Approximation, Gradient-Based Methods, Policy Gradient Methods, REINFORCE Algorithm, Actor-Critic Methods, Introduction to Deep Reinforcement Learning, Deep Q-Networks (DQN), Experience Replay and Target Networks.	8
Module 5: Advanced Topics and Applications Model-Based Reinforcement Learning, Approximate Dynamic Programming, Multi-Agent Reinforcement Learning, Partially Observable Markov Decision Processes (POMDP), Safe and Explainable Reinforcement Learning, Reinforcement Learning in: (Robotics and Autonomous Systems, Recommendation Systems, Resource Management and Scheduling, Smart Grids and Wireless Networks, Game Playing and Industrial Automation), Recent Research Trends and Case Studies	8

Text Books:

1. Sutton, R. S., & Barto, A. G., Reinforcement learning: An introduction (2nd ed.). Cambridge, MA: MIT Press. 2018, ISBN: 978-0262039246.
2. Graesser, L., & Keng, W. L., Foundations of deep reinforcement learning: Theory and practice in Python. Boston, MA: Addison-Wesley Professional. 2019, ISBN: 978-0135172384.
3. Szepesvári, C., Algorithms for reinforcement learning. San Rafael, CA: Morgan & Claypool Publishers, 2010. ISBN: 978-1608454921.

Reference Books:

1. Russell, S. J., & Norvig, P., Artificial intelligence: A modern approach (4th ed., Global ed.). Harlow, England: Pearson Education, 2021, ISBN: 978-1292401133.
2. Lapan, M., Deep reinforcement learning hands-on (2nd ed.). Birmingham, UK: Packt Publishing, 2020, ISBN: 978-1838826994.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of board/LCD projectors/OHPs
CD2	Concept-based Assignments
CD3	Laboratory Experiments/ Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures/Flipped Classroom Activities
CD6	Self-Learning through NPTEL materials and Internet resources
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	1	0	1	1
CO2	3	2	2	2	2	2	2	2
CO3	3	3	3	3	3	2	2	2
CO4	2	3	3	2	3	2	2	2
CO5	2	3	3	2	2	2	2	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4, CD3
CO2	CD1, CD2, CD5
CO3	CD1, CD2, CD5, CD6
CO4	CD1, CD2, CD4, CD5, CD7
CO5	CD1, CD2, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26615

Course Title: Natural Language Processing

Pre-requisite(s): Machine Learning

Co-requisite(s): Machine Learning Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teacher: Dr Indrajit Mukherjee, Dr Komal Naaz

COURSE OBJECTIVES

This course enables the students to:

1. Provide foundational knowledge of Natural Language Processing (NLP), including text processing techniques, linguistic challenges, and methods for preparing natural language data.
2. Develop understanding of statistical and classical NLP approaches for language modeling, text representation, sequence analysis, and syntactic processing.
3. Introduce semantic analysis techniques and lexical representations for understanding and modeling textual information.
4. Enable students to apply machine learning and deep learning approaches for solving Natural Language Processing tasks and developing intelligent language applications.
5. Familiarize students with transformer-based architectures, modern NLP systems, and emerging language technologies.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Apply text preprocessing techniques and foundational NLP concepts to prepare and analyze natural language data.
CO2	Analyze and implement statistical and classical NLP methods for language modeling, text representation, syntactic analysis, and information extraction tasks.
CO3	Employ semantic analysis techniques, lexical resources, and word representations for understanding and modeling textual information.
CO4	Develop NLP solutions using machine learning and deep learning models for tasks such as text classification, sentiment analysis, and sequence processing.
CO5	Develop and Evaluate transformer-based NLP models and modern language applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Natural Language Processing and Text Processing Introduction to Natural Language Processing and its applications, NLP pipeline, challenges in natural language understanding, text preprocessing techniques: tokenization, normalization, stop-word removal, stemming and lemmatization, regular expressions for text processing, spelling correction, ambiguities in NLP	7
Module 2: Statistical and Classical NLP Methods Language Modeling: N-gram models, smoothing techniques, perplexity, Part-of-Speech (POS) Tagging, Hidden Markov Models (HMM), Viterbi Algorithm, Named Entity Recognition (NER), text representation techniques: Bag of Words (BoW), TF-IDF, feature extraction for text **Syntactic Parsing:** Context-Free Grammars (CFGs), CYK Parser **Demonstration**: Information Extraction and classical NLP applications such as Spam filtering, Sentiment classification, Keyword extraction, Information retrieval, Chatbots (rule-based), Document classification	9
Module 3: Semantic Analysis and Text Representation Lexical semantics, lexical resources and WordNet, semantic similarity measures (Cosine similarity), word representations and embeddings: Word2Vec, GloVe, FastText, sentence embeddings (conceptual overview), Word Sense Disambiguation (WSD).	7
Module 4: Machine Learning and Deep Learning for NLP Feature engineering for text, text classification, sentiment analysis, sequence labeling tasks, machine learning approaches for NLP: Logistic Regression and Support Vector Machines (SVM), Introduction to deep learning for NLP, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), attention mechanism, sequence-to-sequence models.	9
Module 5: Transformer Models and Modern NLP Applications	8

Transformer architecture and self-attention mechanism, contextual embeddings, BERT, introduction to Large Language Models (LLMs), Retrieval-Augmented Generation (conceptual overview) **Demonstration:** Demonstration of an NLP application using pre-trained transformer models (such as sentiment analysis, text summarization, question answering, document classification, chatbot, or text generation).	

Text Books:

1. Daniel Jurafsky, James H. Martin, Speech and Language Processing, Pearson Education, 3rd Edition, 2026
2. Christopher D. Manning & Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1st edition, 1999.

Reference Books:

1. Goldberg, Y., Morgan, Neural Network Methods in Natural Language Processing, Claypool Publishers, 1st Edition, 2019.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards
CD2	Assignments/Seminars
CD3	LCD projectors/OHP projectors/Smart Board
CD4	Online Lectures
CD5	Quiz
CD6	Course material (Slides/PDF)
CD7	Mini Projects

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	0	1	0	0	3
CO2	3	3	3	2	2	1	1	2
CO3	3	3	2	3	1	1	2	3
CO4	3	3	3	3	2	1	2	3
CO5	3	3	3	3	2	1	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD3, CD5, CD6
CO2	CD1, CD3, CD5, CD6
CO3	CD1, CD2, CD3, CD4, CD5, CD6
CO4	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD6, CD7

TRACK: Web Application

COURSE INFORMATION SHEET

Course Code: CA26617

Course Title: Modern Web Architecture

Pre-requisite(s): Database Management System, Web Application Development using HTML and CSS

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Abhijit Mustafi, Dr Lopamudra Hota

COURSE OBJECTIVES

This course enables the students to:

1.	Understand the evolution, principles, and architecture of modern web systems.
2.	Design client-side and server-side web applications using modern architectural patterns.
3.	Develop scalable web applications using RESTful services, microservices, and cloud-native technologies.
4.	Apply modern web security, authentication, and performance optimization techniques.
5.	Design, deploy, and manage production-ready web applications using DevOps and cloud deployment practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the principles, architecture, and lifecycle of enterprise web applications.
CO2	Design presentation-layer architectures using MVC, Front Controller, and layered architectural patterns.
CO3	Develop enterprise web architectures using business-layer components, persistence mechanisms, SOA, and RESTful services.

CO4	Apply security mechanisms, middleware, session management, and monitoring techniques to secure and maintain web applications.
CO5	Design scalable, distributed, and cloud-ready web architectures using caching, load balancing, microservices, and emerging architectural paradigms.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Web Application Architecture and Design Evolution of Web Applications, Internet and Web Architecture, Characteristics of Web Applications, Static vs Dynamic Web Applications, Web Application Requirements, Functional and Non-functional Requirements, Enterprise Web Application Architecture, Client–Server Computing,	8
Module 2: Web Framework Architecture and MVC Multi-tier Architecture, Web Request Processing Lifecycle, Web Request Processing Lifecycle, User Interface Architecture, MVC (Model–View–Controller), Front Controller Pattern, View Rendering Technologies, Client-side Scripting Architecture, Single Page vs Multi-page Architectures	8
Module 3: Business Logic and Persistence Architecture Business Logic Layer, Business Components, Transaction Management, Persistence Layer, Data Access Layer, Object Relational Mapping (ORM), Service-Oriented Architecture (SOA), RESTful Web Services	8
Module 4: Enterprise Integration, Security, and Middleware Enterprise Application Integration, Middleware Architecture, Session Management, Authentication and Authorization, Web Security Threats, OWASP Overview, Secure Communication, SSL/TLS, Exception Handling, Monitoring Enterprise Web Applications	8
Module 5: Scalable and Cloud-Native Web Architectures Scalability Principles, Load Balancing, Caching Strategies, Performance Optimization, Distributed Web Architecture, Cloud-based Web Architecture, Microservices and Container-based Deployment, Emerging Trends in Web Architecture	8

Text Books:

1. Leon Shklar, Rich Rosen, Web Application Architecture: Principles, Protocols and Practices, Wiley, 2009

2. Mark Richards, Neal Ford, Fundamentals of Software Architecture, 2nd Edition, O'Reilly, 2025.

Reference Books:

1. Terry Felke-Morris, Web Development & Design Foundations with HTML, 10th Edition, Pearson College Div, 2020.
2. Martin Fowler, Patterns of Enterprise Application Archi, Addison-Wesley, 2002.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and the internet

Code	Course Delivery Method
CD7	Project-based learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	0	2
CO2	3	3	3	2	2	1	0	2
CO3	2	3	3	3	2	2	1	2
CO4	2	3	3	3	2	1	3	3
CO5	2	2	3	3	3	2	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26619

Course Title: Web Security

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): none

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr Prashant Pranav, Dr Ravishankar Mehta

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals of web security and Internet technologies.
2. Study common web application vulnerabilities and their causes.
3. Learn secure authentication, authorization, and session management techniques.
4. Understand encryption, secure configuration, and third-party code security.
5. Develop secure web applications by applying defensive security practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamentals of web security, Internet architecture, browsers, and web servers.
CO2	Analyze common web application vulnerabilities and recommend suitable mitigation techniques.
CO3	Apply secure authentication, session management, authorization, and information protection mechanisms.
CO4	Implement secure communication, encryption, dependency management, and XML security practices.
CO5	Evaluate advanced web security threats and apply best practices for building secure web applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Web Security Introduction to Web Security: Security Principles, Threat Landscape, and Secure Development Lifecycle. Internet Architecture: TCP/IP Protocol Suite, Application Layer Protocols, Stateful Communication, and Secure Data Transmission. Web Browser Architecture: Rendering Engine, JavaScript Execution Environment, Browser Security Model, and Client-Side Processing. Web Server Architecture: Static and Dynamic Content Delivery, Client–Server Interaction, and Request Processing. Software Development Practices: Requirements Analysis, Secure Design, Implementation, Testing, Deployment, Maintenance, and Dependency Management.	8
Module 2: Web Application Vulnerabilities Injection Vulnerabilities: SQL Injection, Command Injection, Code Injection, File Inclusion, and Unrestricted File Uploads. Cross-Site Scripting (XSS): Stored, Reflected, and DOM-Based XSS, Attack Scenarios, and Mitigation Strategies. Cross-Site Request Forgery (CSRF): Attack Mechanisms, RESTful Design Principles, Anti-CSRF Tokens, SameSite Cookies, Reauthentication, and Secure Request Validation.	8
Module 3: Authentication and Access Security Authentication Mechanisms: Password-Based Authentication, Multi-Factor Authentication, Single Sign-On, and Federated Identity Management. Session Management: Session Creation, Secure Cookie Handling, Session Hijacking, and Session Protection. Authorization and Access Control: Privilege Escalation, Role-Based Access Control, Directory Traversal Prevention, Information Disclosure, Secure Error Handling, and Security Advisory Management.	8
Module 4: Secure Infrastructure Cryptography and Secure Communication: HTTPS, TLS, Certificates, and Secure Transmission. Secure Configuration Management: Hardening Servers, Managing Dependencies, Protecting Third-Party Services, and Configuration Best Practices. XML Security: XML Validation, XML External Entity (XXE) Prevention, XML Bombs, and Secure XML Processing.	8
Module 5: Advanced Security Practices Advanced Web Security Threats: Phishing and Email-Based Attacks, Malicious URL Obfuscation, Clickjacking, Server-Side Request Forgery (SSRF), Botnets, and Distributed Denial-of-Service (DDoS) Attacks. Security Monitoring and Incident Response: Threat Detection, Risk Mitigation, Secure Coding Best Practices, and Emerging Trends in Web Application Security.	8

Text Books:

1. McDonald, M., Web security for developers: Real threats, practical defense. No Starch Press, 2020.

Reference Books:

1. Hoffman, A., Web application security: Exploitation and countermeasures for modern web applications. O'Reilly Media, 2020.
2. Stuttard, D., & Pinto, M., The web application hacker's handbook: Finding and exploiting security flaws (2nd ed.). Wiley, 2011.
3. Sullivan, B., & Liu, V., Web application security: A beginner's guide. McGraw-Hill Education, 2011.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Interactive classroom lectures using presentations and board work
CD2	Tutorial sessions for mathematical problem solving and security algorithm analysis

Code	Course Delivery Method
CD3	Demonstration of security protocols, and network security tools
CD4	Assignment-based learning involving analytical and numerical problems
CD5	Case studies on cryptographic applications and network security incidents
CD6	Student seminars and presentations on emerging trends in web security and cybersecurity
CD7	Self-learning through textbook reading, research articles, and online learning resources

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	0	1	0	0	0	1
CO2	2	3	2	2	0	0	2	1
CO3	2	2	3	2	0	0	3	1
CO4	2	2	3	3	0	1	3	2
CO5	2	3	3	2	1	1	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD2, CD4, CD7
CO3	CD1, CD2, CD5, CD6
CO4	CD6, CD7
CO5	CD1, CD2, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26621

Course Title: Frontend Design 2

Pre-requisite(s): Frontend Design 1, Web Application Development using HTML and CSS

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Abhijit Mustafi, Dr. Sandip Ghosal

COURSE OBJECTIVES

This course enables the students to:

1.	Understand fundamental JavaScript programming constructs, including variables, arrays, operators, expressions, control flow, and functions, to build interactive scripts.
2.	Control browser behavior and web page content dynamically using the Window object and the Document Object Model (DOM).
3.	Implement event-driven programming, form handling, and dynamic styling using JavaScript and CSS.
4.	Develop asynchronous, data-driven web applications using AJAX, JSON, and regular expressions.
5.	Build modern, feature-rich web applications using HTML5 APIs such as local storage, geolocation, and canvas.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Write JavaScript programs using variables, arrays, operators, control structures, and functions to solve basic programming problems. (BL3)
CO2	Apply DOM and Window object methods to control browser behavior and manipulate web page content dynamically. (BL3)
CO3	Implement event handling, form validation, and CSS manipulation using JavaScript to build interactive web pages. (BL3)
CO4	Develop AJAX-based applications that exchange JSON data and use regular expressions for pattern matching and validation. (BL3)
CO5	Design and build HTML5-based web applications using local storage, geolocation, canvas, and polyfills for cross-browser compatibility. (BL6)

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: JavaScript Fundamentals Introduction to JavaScript, Variables, Arrays, Operators, Expressions, Statements, Flow control, Iterative statements, Writing functions, Creating objects.	4
Module 2: JavaScript DOM and Browser Programming Javascript on the web, Controlling the browser with the Window object, Using the DOM	10
Module 3: Event-Driven Web Programming Events and Event handling in Javascript, Form handling, working with CSS in Javascript	8
Module 4: Asynchronous JavaScript and Data Processing Working with AJAX and JSON in Javascript, Using regular expressions	9
Module 5: HTML5 APIs and Modern Web Features HTML5 API, local storage, geolocation, Drawing shapes with canvas, shims and polyfills	9

Text Books:

1. Jones D., Javascript – Novice to Ninja, 2nd Edition, Site Point, 2017

Reference Books:

1. Minnick C., Hooland E., Coding with Javascript for Dummies, Wiley, 2015

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	0	1
CO2	2	2	2	2	1	0	0	1
CO3	2	2	2	2	1	0	1	1
CO4	2	3	3	2	1	0	1	1
CO5	2	2	3	3	1	0	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD3, CD5
CO4	CD1, CD2, CD4, CD5
CO5	CD1, CD4, CD5, CD7

TRACK: Advanced Modern Computing

COURSE INFORMATION SHEET

Course Code: CA26623

Course Title: Advanced Database Systems

Pre-requisite(s): Database Management Systems

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. V. K. Jha, Dr. S. Pushkar, Dr. O. Tigga

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts of transaction processing, including transaction properties, scheduling, recoverability, serializability, and SQL transaction support.
2. Analyze and compare various concurrency control techniques used to ensure correctness, consistency, and performance in multi-user database systems.
3. Apply database recovery mechanisms and backup strategies to maintain data integrity and ensure fault tolerance under different failure scenarios.
4. Examine the principles, architecture, and transaction management techniques of distributed database systems, including data fragmentation, replication, and distributed concurrency control.
5. Evaluate database security mechanisms and privacy-preserving techniques to protect database systems against unauthorized access, SQL injection, and other security threats.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain transaction processing concepts and scheduling techniques.
CO2	Analyze concurrency control methods for database consistency.
CO3	Apply database recovery and backup techniques.

CO4	Analyze distributed database design and transaction management.
CO5	Evaluate database security and privacy protection mechanisms.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Database Transaction Management Transaction Processing Concepts and Theory: Review of basic concepts, Transaction and System Concepts, Desirable Properties of Transactions, Characterizing Schedules Based on Recoverability, Characterizing Schedules Based on Serializability, Transaction Support in SQL.	8
Module 2: Concurrency Control Techniques Concurrency Control Techniques: Two-Phase Locking Techniques for Concurrency Control, Concurrency Control Based on Timestamp Ordering, Multiversion Concurrency Control Techniques, Validation (Optimistic) Concurrency Control Techniques, Granularity of Data Items and Multiple Granularity Locking.	8
Module 3: Database Recovery and Fault Tolerance Recovery Concepts, NO-UNDO/REDO Recovery Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, Recovery in Multidatabase Systems, Database Backup and Recovery from Catastrophic Failures.	8
Module 4: Distributed Database Concepts Distributed Databases and Client-Server Architectures, Distributed Database Concepts, Data Fragmentation, Replication and Allocation Techniques for Distributed Database Design, Concurrency Control and Recovery in Distributed Databases, Transaction Management in Distributed Databases, Types of Distributed Database Systems, General Architecture of Pure Distributed Databases.	8
Module 5: Recovery System Database Security: Introduction to Database Security Issues, Discretionary Access Control Based on Granting and Revoking Privileges, SQL Injection, Encryption and Public Key Infrastructures, Privacy Issues and Preservation.	8

Text Books:

1. Elmasri Ramez, & Navathe S.B., Fundamentals of Database Systems, 5th Edition, Pearson Education, 2013.

Reference Books:

1. C.J. Date, "An introduction to Database Systems", 7th Edition.,Pearson Education, New Delhi, 2004.
2. H. Korth et al., "Database Management System Concepts", 3rd Edition, TMH, New Delhi, 2002.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects

Code	Course Delivery Method
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	1	2	0	1
CO2	3	3	2	2	1	1	1	1
CO3	2	2	3	3	2	2	2	1
CO4	3	3	3	3	2	2	1	2
CO5	2	3	2	2	2	1	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5, CD6
CO2	CD1, CD2, CD3, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD5, CD6
CO4	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD3, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26625

Course Title: Bioinformatics

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Shripal Vijayvargiya, Dr. Amritanjali, Dr. Kanchan Jha

COURSE OBJECTIVES

The course aims to enable students to:

1. Understand the role of Computer Science in solving biological and healthcare problems.
2. Learn biological concepts necessary for computational analysis of biological sequences.
3. Apply algorithms, databases, machine learning, and data mining techniques to biological datasets.
4. Use modern bioinformatics software for sequence analysis and biological data processing.
5. Explore applications in genomics, drug discovery, healthcare, and precision medicine.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain fundamental biological concepts and organize biological information using computational databases.
CO2	Apply string matching, dynamic programming, and database searching algorithms to analyze DNA and protein sequences.
CO3	Apply algorithms to analyze multiple sequence alignment and efficiently search the genomic and proteomic databases.
CO4	Develop computational solutions for genomic, proteomic, and evolutionary data analysis using statistical and machine learning techniques.
CO5	Analyze biological networks, protein structures, and molecular interactions using computational models and visualization tools.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction What is Bioinformatics? Computational Biology vs Bioinformatics, Applications in Healthcare, Agriculture, and Pharmaceuticals. Basic Biology Concepts: Cell structure, DNA, RNA and Protein, Central Dogma of Molecular Biology, Genes and Chromosomes, Genetic Code. Biological Data: Types of biological data, DNA, RNA and Protein sequences, Bioinformatics Resources and Databases: GenBank, EMBL, DDBJ, UniProt, PDB, and other databases, organization of databases: data contents, purpose and utility. Sequence formats (FASTA, GenBank).	7
Module 2: Algorithms for Biological Sequence Analysis Meaning of sequence, sequence similarity, homology, meaning of alignment. String Matching Algorithms: Exact matching, Approximate smatching, Edit distance, Longest Common Subsequence. Pairwise Sequence Alignment: Different scoring models, Substitution matrices (PAM and BLOSUM), Pairwise Alignment: Concept of Global and Local Alignment, Dot matrix method, Dynamic programming (NeedlemanWunsch algorithm, Smith-Waterman algorithm, Choosing of best scoring matrix, gap penalties, Significance of score, FASTA and BLAST algorithms.	9
Module 3: Multiple Sequence alignment Introduction, Applications, Phylogenetic analysis, Multiple Sequence Alignment methods, (MSA), Scoring of a MSA, Progressive (CLUSTALW and PILEUP), Iterative (Genetic) and Hidden Markov Model (HMM) methods of MSA, Local MSA (Profile and BLOCK analysis, and Pattern searching, and Expectation Maximization (EM) Algorithm (MEME) and Gibbs Sampler.	8
Module 4: Computational Genomics and Machine Learning Genome assembly and annotation, Comparative genomics, Gene prediction algorithms, Transcriptomics and RNA-Seq analysis, Feature engineering for biological datasets, Machine Learning in Bioinformatics, Classification of gene expression data, Disease prediction, Protein function prediction, Clustering algorithms for gene expression analysis.	8
Module 5: Structural Bioinformatics and Computational Biology Protein Structure: Primary, Secondary, Tertiary and Quaternary structures, Protein folding. Protein Structure Prediction: Homology modelling (overview), Introduction of AlphaFold, Molecular Visualization. Molecular Docking: Drug-target interaction, Virtual screening, Drug discovery pipeline,	8

Introduction to Protein interaction networks and Gene regulatory networks.	

Text Books:

1. David W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, 2024.
2. Arthur M. Lesk, Introduction to Bioinformatics, Oxford University Press, 2019.

Reference Books:

1. Andreas D. Baxevanis and B. F. F. Ouellette, Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Wiley, 2004.
2. Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, 1998.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Chalk and Board / Traditional Lecture Method

Code	Course Delivery Method
CD2	Power Point Presentations (PPT)
CD3	E-Learning (NPTEL, Coursera, YouTube)
CD4	Problem Solving / Tutorial Sessions
CD5	Case Study Based Learning
CD6	Practical / Laboratory / Simulation Tools (MATLAB, Python)
CD7	Group Discussion / Seminar

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	1	2	1	0	0	1
CO2	3	3	2	2	2	0	0	1
CO3	3	3	3	3	2	1	0	2
CO4	3	3	3	3	3	2	1	2
CO5	3	3	3	3	3	2	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD3, CD4, CD6
CO4	CD1, CD2, CD3, CD4, CD5, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26627

Course Title: Retrieval Augmented Generation

Pre-requisite(s): Python Programming, Natural Language Processing

Co-requisite(s): None

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr Debjani Mustafi, Dr Aditi Panda

COURSE OBJECTIVES

This course enables the students to:

1.	Understand the fundamentals of Retrieval-Augmented Generation (RAG) and information retrieval.
2.	Learn document processing, embeddings, and vector databases used in RAG systems.
3.	Develop simple Retrieval-Augmented Generation pipelines using modern frameworks.
4.	Explore advanced retrieval techniques and methods for improving response quality.
5.	Identify applications and evaluation methods for Retrieval-Augmented Generation systems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the concepts of information retrieval, vector representations, and Retrieval-Augmented Generation.
CO2	Apply document processing, embedding models, and vector databases for knowledge retrieval.
CO3	Develop basic Retrieval-Augmented Generation pipelines using modern frameworks and large language models.
CO4	Analyze retrieval strategies and techniques for improving the performance of RAG systems.
CO5	Evaluate Retrieval-Augmented Generation systems for different application domains using appropriate performance metrics.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Retrieval-Augmented Generation Introduction to Large Language Models and limitations (hallucinations, knowledge cutoff), Fundamentals of Information Retrieval, Sparse Retrieval: TF-IDF and BM25, Dense Retrieval: Embeddings and Semantic Search, RAG Architecture and Workflow.	8
Module 2: Building the Knowledge Base Data sources and document ingestion, Text preprocessing and document parsing, Document chunking strategies, Embedding models and vector representations, Vector databases and indexing (FAISS, Chroma, Pinecone, Weaviate).	8
Module 3: Retrieval and Generation Pipelines Query processing and embedding generation, Similarity search and nearest-neighbor retrieval, Prompt engineering for RAG, Building end-to-end RAG pipelines using LangChain/LlamaIndex, Multi-document retrieval and response synthesis	8
Module 4: Advanced Retrieval Techniques Hybrid Retrieval, Query Transformation and Expansion, Reranking using Cross Encoders and ColBERT, Agentic RAG, Adaptive RAG, Hierarchical RAG	8
Module 5: Advanced RAG Systems and Applications Graph RAG and Knowledge Graph Integration, Multimodal RAG, Evaluation Metrics for Retrieval and Generation, Explainability and Trustworthy RAG, Security and Privacy in RAG Systems, Case Studies and Applications	8

Text Books:

1. Abhinav Kimothi, A Simple Guide to Retrieval Augmented Generation, Manning Publications, 2025.
2. Deepak Dhyani, RAG with Python Cookbook: Learn Principles of RAG with LLM and Agentic AI, BPB Publications, 2025

Reference Books:

1. Ben Auffarth, Retrieval Augmented Generation: The Seminal Papers, Manning Publications, 2025.
2. Bahaaldine Azarmi, RAG-Driven Generative AI, Packt Publishing, 2025.
3. Hands-On RAG for Production, O'Reilly Media, 2025.
4. Mastering Retrieval-Augmented Generation: Advanced Techniques for Enterprise AI Systems, O'Reilly Media, 2025.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Interactive Lectures
CD2	Demonstration of RAG Frameworks and Tools
CD3	Problem Solving Sessions
CD4	Case Study Discussions
CD5	Seminar and Student Presentations
CD6	Assignment-Based Learning
CD7	Project-Based Learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	3	2	2
CO2	3	2	3	2	2	2	1	1
CO3	3	3	3	2	3	3	3	2
CO4	2	3	3	3	2	2	2	2
CO5	3	2	2	2	3	3	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD3, CD4
CO3	CD3, CD4, CD6, CD7
CO4	CD2, CD4, CD5, CD6
CO5	CD2, CD5, CD7

PROGRAM ELECTIVE III

TRACK: CORE COURSES

COURSE INFORMATION SHEET

Course Code: CA26629

Course Title: Internet of Things

Pre-requisite(s): Data Communication and Computer Networks

Co-requisite(s): Embedded System Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Itu Snigdh, Dr. Umesh Prasad

COURSE OBJECTIVES

This course enables the students to:

1. Understand the basic concept and the IoT Paradigm.
2. Know the state of art architecture for IoT applications.
3. Learn the available protocols used for IoT.
4. Design basic IoT Applications.
5. Evaluate optimal IoT applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Identify the IoT Components and its capabilities.
CO2	Explain the architectural view of IoT under real world constraints.
CO3	Analyse the different Network and link layer protocols.
CO4	Evaluate and choose among the transport layer protocols.
CO5	Design an IoT application.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management.	8
Module 2: IoT Architecture - State of the Art Introduction, State of the art, Reference Model and architecture, IoT reference Model - IoT Reference Architecture.	8
Module 3: IoT Data Link Layer & Network Layer Protocols PHY/MAC Layer (3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART, Bluetooth Low Energy, Zigbee Smart Energy, Network Layer-IPv4, IPv6, 6LoWPAN	8
Module 4: Transport & Session Layer Protocols Transport Layer: TCP, MPTCP, UDP, Session Layer: HTTP, CoAP, XMPP, AMQP, MQTT.	8
Module 5: Layer Protocols & Security Service Layer -oneM2M, ETSI M2M, security in IoT and M2M applications.	8

Text Books:

1. Holler Jan, Tsiatsis Vlasios, Mulligan Catherine, Avesand Stefan, Karnouskos Stamatis, Boyle David, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st Edition, Academic Press, 2014.
2. Waher Peter, Learning Internet of Things, PACKT publishing, BIRMINGHAM- MUMBAI, 2015

Reference Books:

1. Reiter Bernd Scholz, Michahelles Florian, Architecting the Internet of Things, Springer, 2011, ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2.
2. Minoli Daniel, Building the Internet of Things with IPv6 and MIPv6, Wiley, 2013.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	3	3	2	1	0	3	3
CO2	3	3	3	3	1	0	3	3
CO3	3	3	3	3	1	0	3	3
CO4	3	3	3	3	1	0	3	3
CO5	3	3	3	3	1	0	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26631

Course Title: Shell Scripting

Pre-requisite(s): Python using Linux

Co-requisite(s): Shell Scripting Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. S. Kanungo, Dr. Md. S. Fahad

COURSE OBJECTIVES

This course enables the students to:

1. Understand Linux OS architecture, the Bash environment, and core commands for file and system management
2. Develop functional shell scripts using procedural programming constructs to solve system and computational problems
3. Apply regular expressions, sed, and gawk utilities for efficient text processing, file manipulation, and report generation.
4. Design modular and reusable shell scripts to automate system administration tasks, including process management, backup, scheduling, and user account management.
5. Implement shell scripting solutions for automating system-level operations and developing practical Linux-based utilities.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Demonstrate proficiency in Linux commands and shell scripting fundamentals.
CO2	Develop shell scripts using decision-making, looping constructs, and command-line arguments.
CO3	Design modular shell programs using functions, regular expressions, sed, and gawk.
CO4	Automate system administration, file processing, and text manipulation tasks using shell scripts
CO5	Develop automation using Shell Scripting for system-level applications

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Linux Shell Fundamentals and Shell commands Introduction to Linux System & History, Features of Linux, Advantages and disadvantages of Linux, Introduction to File System, Memory Management, Shell and its Types, parent-clild shell relatioship, Basic Commands: login, logout, date, man, pwd, who, whoami, dir, ls, cd, mkdir, rmdir, cp, mv, rm, rev, file redirection, cut, paste, sort, File permission and use of chmod command, Use of Wild card characters, Introduction to vi editor. Introduction to environment variable like HOME, PATH.	8
Module 2: Shell Programming Introduction to shell script: execution of shell script, variables, pipes, expr, bc and test commands, Control structure: if, if..else, case structure, Iteration: while, for construct, break, continue, exit commands, User-defined functions, Recursive functions, Arrays in shell scripting, CASE in shell scripting, Passing Parameters-Tracking parameters, Handling Signals, running scripts in the background, controlling a Job, modifying script priority, Automating script execution.	10
Module 3: File Processing and Text Manipulation Regular expressions, Pattern matching, Introduction to grep and sed, Introduction to AWK, program structure, pattern-action model, variables, operators, built-in functions, formatted output, and file processing.	6
Module 4: System Automation and Shell Utilities Shell features and Portability, Shell scripting for system administration, Shell scripting for automation, Shell scripting for database interaction, Network and Internet utilities in shell scripting.	8
Module 5: Kernel Interfacing, Systems Telemetry, and Process Lifecycle The Virtual Filesystem & Kernel Abstraction Layers, Parsing Hardware & Memory Telemetry, Low-Level Process Auditing via Shell, Runtime Kernel Tuning & Configuration (sysctl), Kernel Ring Buffer Analytics & Hardware Events, Kernel Module Orchestration.	8

Text Books:

1. Richard Blum, Christine Bresnahan, Linux Command Line and Shell Scripting BIBLE, Wiley Publishing, 3rd Edition, 2015.

2. Robert Love, Linux System Programming, O'Reilly Media, Inc, 2013

Reference Books:

1. Clif Flynt, Sarath Lakshman, Shantanu Tushar, Linux Shell Scripting Cookbook, Packt Publishing, 3rd Edition, 2017.
2. Stephen G. Kochan, Patrick Wood, Shell Programming in Unix, Linux, and OS X, Addison Wesley Professional, 4th Edition, 2016.
3. Mokhtar Ebrahim, Andrew Mallett, Mastering Linux Shell Scripting, Packt Publishing, 2nd Edition, 2018.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities

Code	Course Delivery Method
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	2	0	2	3	3
CO2	3	2	2	2	0	2	0	2
CO3	3	3	3	3	2	3	1	3
CO4	3	3	3	3	2	3	2	3
CO5	3	3	3	3	2	3	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5, CD6
CO2	CD1, CD2, CD3, CD4, CD5, CD6
CO3	CD1, CD2, CD3, CD4, CD5, CD6
CO4	CD1, CD2, CD3, CD4
CO5	CD1, CD2, CD3, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26633

Course Title: Mobile Application Development using Flutter

Pre-requisite(s): Object-Oriented Programming, Database Management Systems, Web Technologies

Co-requisite(s): Mobile Application Development using Flutter Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Jit Mukherjee, Dr. I. Mukherjee

COURSE OBJECTIVES

This course enables the students to:

1. Understand the architecture of modern cross-platform mobile application development.
2. Develop scalable mobile applications using Flutter and Dart.
3. Design responsive, reusable, and maintainable user interfaces.
4. Integrate mobile applications with cloud services, databases, and RESTful APIs.
5. Apply software engineering principles, testing methodologies, and deployment practices for mobile applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain Flutter architecture, Dart language features, and cross-platform development concepts.
CO2	Design responsive mobile interfaces using advanced Flutter widgets and state management techniques.
CO3	Develop secure mobile applications integrating cloud databases, REST APIs, and device services.
CO4	Evaluate and optimize Flutter applications for performance, scalability, testing, and security.
CO5	Design, develop, deploy, and document an industry-oriented mobile application.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Flutter Framework and Advanced Dart Mobile Application Development Trends, Cross-platform Development Frameworks, Flutter Architecture, Rendering Engine, Widget Tree, Flutter SDK and Project Structure, Dart Language Features, Null Safety, Functional Programming, Collections, Futures, Streams, Asynchronous Programming	8
Module 2: User Interface Design and State Management Material Design, Cupertino Widgets, Responsive UI, Adaptive Layouts, Theme Management, Forms and Validation, Navigation, Named Routes, Custom Widgets, State Management – Provider, Riverpod, BLoC Architecture, GetX; Animations, Localization and Internationalization	8
Module 3: Mobile Backend Integration RESTful Web Services, HTTP and Dio Packages, JSON Serialization, GraphQL (Overview), Firebase Platform, Firebase Authentication, Cloud Firestore, Firebase Storage, Shared Preferences, SQLite, Hive Database, Secure Storage, Offline Data Synchronization	8
Module 4: Advanced Mobile Services and Security Camera Integration, Image Processing, Google Maps, GPS and Geolocation, QR Code and Barcode Scanner, Push Notifications, Device Sensors, Permissions Management, Mobile Security, OAuth Authentication, JWT Authentication, Secure Coding Practices, Flutter Performance Optimization, Accessibility Guidelines	8
Module 5: Testing, DevOps and Application Deployment Unit Testing, Widget Testing, Integration Testing, Debugging Tools, Performance Profiling, Code Quality and Static Analysis, Git and Version Control, Continuous Integration and Continuous Deployment (CI/CD), Android APK and App Bundle Generation, Play Store Deployment, Introduction to iOS Deployment, Flutter Web and Desktop Applications, AI Service Integration using REST APIs (e.g., LLM-based services)	8

Text Books:

1. Eric Windmill, Flutter in Action, Manning Publications, 2020.
2. Carmine Zaccagnino, Programming Flutter: Native, Cross-Platform Apps the Easy Way - Pragmatic Bookshelf, 2020.

3. Marco L. Napoli, Beginning Flutter: A Hands-On Guide to App Development, Wiley, 2019.
4. Fu Cheng, Flutter Recipes, Apress, 2019.

Reference Books:

1. Alessandro Biessek, Flutter Complete Reference, BPB, 2021
2. Priyanka Tyagi, Pragmatic Flutter, CRC Press, 2022
3. Raj Kamal, Mobile Computing, Oxford University Press, 2019.
4. Bill Phillips, Chris Stewart, and Kristin Marsicano, Android Programming: The Big Nerd Ranch Guide. Pearson Education, 2019.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects

Code	Course Delivery Method
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	1	1	1	1	1	1	1
CO2	2	1	2	2	1	1	1	1
CO3	2	1	2	3	1	1	1	1
CO4	3	2	3	3	1	2	1	1
CO5	3	2	3	3	1	2	1	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD3, CD4, CD5, CD7
CO2	CD1, CD3, CD4, CD5, CD7
CO3	CD1, CD3, CD4, CD5, CD7
CO4	CD1, CD3, CD4, CD5, CD7
CO5	CD1, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26635

Course Title: Software Testing and Version Control

Pre-requisite(s): Software Engineering, Software Design Pattern

Co-requisite(s): Software Testing and Version Control Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. K. Sridhar Patnaik, Dr. K. Rajnish

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals, principles, and importance of software testing in the Software Development Life Cycle (SDLC).
2. Apply various software testing techniques and methodologies to identify defects and improve software quality.
3. Analyze software requirements and design effective test plans, test cases, and test reports.
4. Utilize automated testing tools and frameworks to enhance testing efficiency and reliability.
5. Employ version control systems and collaborative development practices for managing software projects effectively.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain software quality concepts, testing principles, testing levels, and testing life cycle processes used in software development. L2
CO2	Design and execute test cases using black-box and white-box testing techniques to validate software functionality and performance. L3
CO3	Analyze software requirements and defects to develop comprehensive test plans, test scenarios, and defect reports. L4
CO4	Use automated testing frameworks and tools to perform functional and regression testing for software applications. L3
CO5	Apply Git-based version control practices, branching strategies, merging, and collaborative workflows for software project management and team development. L3

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Fundamentals of Software Testing Introduction to Software Quality, Software Development Life Cycle (SDLC) and Software Testing Life Cycle (STLC), Verification vs Validation, Principles of Software Testing, Testing Process and Defect Life Cycle, Levels of Testing -Unit Testing, Integration Testing, System Testing, Acceptance Testing, Functional and Non-functional Testing, Software Quality Standards (ISO/IEC 25010 overview) **Practical Exposure** • Writing Test Scenarios • Defect Reporting using Bug Tracking Template	8
Module 2: Test Design Techniques and Automation Black Box Testing: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition Testing, White Box Testing: Statement Coverage, Branch Coverage, Path Coverage, Grey Box Testing, Regression Testing, Smoke and Sanity Testing, Introduction to Test Automation, Unit Testing Frameworks: JUnit, PyTest (Overview) **Practical Exposure** • Developing Unit Test Cases • Measuring Code Coverage	8
Module 3: Advanced Software Testing: Integration Testing Strategies: System Testing, Performance Testing, Load Testing, Stress Testing, Security Testing, Usability Testing, Test Planning, Test Case Design, Test Execution and Test Reports **Practical Exposure** • Preparing Test Plan • Test Case Documentation • Test Report Preparation	8
Module 4: Version Control using Git and GitHub Need for Version Control: Centralized vs Distributed Version Control, Git Architecture, Repository Creation, Git Installation and Configuration, Basic Git Commands: init, clone, add, commit, status, log. Branching and Merging, Merge	8

Conflicts, GitHub Repository Management Pull Requests, Code Review, Git Ignore, Tags and Releases **Practical Exposure** • Create Git Repository • Branch Management • Collaborative Development using GitHub	
Module 5: Collaborative Software Development and DevOps Basics Software Configuration Management, Continuous Integration, Continuous Delivery (Overview), GitHub Actions (Introduction), Issue Tracking, Agile Testing, Scrum Testing Practices, Test Automation in CI/CD, Software Documentation, Open Source Contribution Workflow, Case Studies from Industrial Projects. **Practical Exposure** • GitHub Collaboration • Pull Request Review • CI Workflow Demonstration • Mini Team Project	8

Text Books:

1. Roger S. Pressman & Bruce Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill, 2023.
2. Glenford J. Myers, Corey Sandler & Tom Badgett, The Art of Software Testing, 3rd Edition, Manohar publishers & distributors, 2010.

Reference Books:

1. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015.
2. Ilene Burnstein, Practical Software Testing, Springer, 2010
3. Boris Beizer, Software Testing Techniques, Wiley India, 2002.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Interactive Lectures
CD2	Tutorial and Problem-Solving Sessions
CD3	Demonstrations
CD4	Project-Based Learning
CD5	Case Study Discussions
CD6	Self-Learning through Online Resources
CD7	Seminars, Presentations, and Peer Learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	0	0	0	0	1	1
CO2	2	3	2	1	0	0	0	0
CO3	2	3	2	1	0	0	1	0
CO4	1	2	2	3	0	0	1	2
CO5	1	1	2	3	3	2	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD5, CD6
CO2	CD1, CD2
CO3	CD2, CD3, CD4, CD5, CD7
CO4	CD3, CD4, CD6, CD7
CO5	CD3, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26637

Course Title: Advanced Programming in Java

Pre-requisite(s): Understanding OOP using Java

Co-requisite(s): Advanced Programming in Java Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teacher: Dr. Jamimamul Bakas, Dr. Siba Mitra, Dr. Richa

COURSE OBJECTIVES

This course enables the students to:

1. To provide advanced knowledge of Java programming concepts and object-oriented application development.
2. To develop the ability to design concurrent applications using multithreading and collections frameworks.
3. To impart skills for developing database-driven applications using JDBC.
4. To enable the development of dynamic web applications using Servlets and JSP technologies.
5. To introduce network programming concepts for developing distributed and client-server applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Apply advanced Java programming concepts and collections framework to develop efficient software applications.
CO2	Develop multithreaded Java applications using synchronization and concurrent programming techniques.
CO3	Construct database-driven applications using JDBC and transaction management.
CO4	Design and implement dynamic web applications using Servlets, JSP, and session management techniques.
CO5	Develop client-server applications using Java networking APIs and integrate them into distributed systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations Review & Essential Core Extensions Generics: Type parameters, parameterized types, bounded wildcards. Collection Framework: Dynamic data manipulation using List, Set, Map, implementations of ArrayList, HashSet, HashMap. Robust Exception Handling and Assertions: Throwing custom exceptions, Basic Idea of Assertions, Purpose of Assertions, Types of Assertions, Assertions versus Exception handlingava 8+ Additions: Lambda e. Jxpressions, functional interfaces, stream API basics.	8
Module 2: GUI programming Components and Facilities or Rich Graphical User Interfaces: Programming with the JFC. Swing API Components, Jcomponent Class, Windows, Dialog Boxes, and Panels, Labels, Buttons, and Check Boxes, Menus, Toolbars, and Actions, Sliders, Spinners, Progress Bars, and Scrollbars, Lists and Combo Boxes, Text-Entry Components, Color and File Choosers, Tables and Trees, Printing with the 2D API, Java Print Service API.	8
Module 3: Database Connectivity (JDBC) JDBC Architecture: Working with the Core java.sql and javax.sql packages. Driver Technologies: Configuring JDBC drivers and establishing database connections. SQL Statements: Executing queries via Statement, PreparedStatement, and CallableStatement. Data Retrieval: Navigating and updating data using the ResultSet metadata.	8
Module 4: Enterprise Web Components (Servlets & JSP) Servlet Life Cycle: Managing initialization, service delivery, and destruction phases. HTTP Communication: Processing client-side requests and generating dynamic responses. Session Tracking: Managing persistent user states using Cookies, Hidden Form Fields, and HttpSession. JavaServer Pages (JSP): Creating presentation layers using scripting tags, directives, and actions.	8
Module 5: Distributed Computing & Network Programming Socket Programming: Building client-server architecture using TCP/IP Sockets. Remote Method Invocation (RMI): Designing distributed applications via RMI architecture, stubs, and skeletons. Networking Tools: Utilizing URL, URLConnection, and InetAddress APIs.	8

Text Books:

1. Roy, Uttam K. Advanced Java Programming. 1st ed., Oxford University Press, 2015.

2. Schildt, Herbert. Java: The Complete Reference. 13th ed., McGraw-Hill Education, 2025.

Reference Books:

1. Programming with Java, Beginner to Advanced, Cay S. Horstmann, 7th Edition, Wiley
2. Java How to Program, Deitel P., Deitel H., Pearson Publications, 2016.
3. Seth, Anita, and B. L. Juneja, Java: One Step Ahead, Oxford University Press, New Delhi, 2017
4. Sierra, Kathy, and Bert Bates. Head First Java. 3rd ed., O'Reilly Media, 2022

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes) (15 marks each)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Tutorial / Assignments
CD3	Self Learning through NPTEL / online materials
CD4	Coding demonstration in class
CD5	Mini projects

Code	Course Delivery Method
CD6	Interactive Lectures, Flipped Classroom Activities
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	2	1	1	1
CO2	3	3	2	1	2	1	1	1
CO3	2	3	2	1	2	1	1	1
CO4	2	3	2	1	2	1	2	1
CO5	2	3	2	1	2	1	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD4
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD4, CD6
CO4	CD1, CD2, CD3, CD4, CD5
CO5	CD1, CD2, CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26630

Course Title: Embedded System Lab

Pre-requisite(s): Python Programming

Co-requisite(s): Internet of Things

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Prof. Abhijit Mustafi, Dr. Aditi Panda

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals, architecture, components and applications of embedded systems and Arduino and/or Raspberry Pi Pico based microcontroller platforms.
2. Develop embedded programs using digital I/O and register-level programming techniques.
3. Interface digital and analog input/output devices using Arduino and/or Raspberry Pi Pico and implement sensor-based applications.
4. Apply timers, PWM, interrupts and serial communication in embedded system development.
5. Design simulation-based embedded solutions for real-world monitoring and control applications using sensors and actuators.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the architecture, operation and applications of embedded systems, Arduino/Raspberry Pi Pico platforms.
CO2	Develop embedded programs using Arduino/Raspberry Pi Pico and register-level programming.
CO3	Interface digital and analog devices using GPIO, ADC and sensors.
CO4	Implement embedded applications using timers, PWM, interrupts and serial communication.
CO5	Design embedded solutions for real-world monitoring and control applications.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Embedded Systems and Arduino Introduction to Embedded Systems: Definition, characteristics and applications, Components of an Embedded System, Microprocessors vs. Microcontrollers, Arduino Uno (ATmega328) and/or Raspberry Pi Pico (RP2040) architecture and overview, Board: Digital, Analog, PWM and Power pins, Breadboards, LEDs, resistors and switches, Simulation-based embedded system development using Tinkercad/Wokwi	9
Module 2: Digital I/O and Register-Level Programming General Purpose Input/Output (GPIO), GPIO architecture of ATmega328P and/or RP2040 (PORTB, PORTC, PORTD), GPIO programming using registers (ATmega328P) and/or GPIO API concepts (RP2040), Bitwise operators and masking techniques, LED interfacing and pattern generation, Software timing and delay generation.	9
Module 3: Analog Interfacing and Sensor Integration Analog vs. Digital signals, Analog-to-Digital Conversion (ADC), Analog input using potentiometer, Push button interfacing, Pull-up and Pull-down resistors, Sensor interfacing using Arduino and/or Raspberry Pi Pico, Thresholding, hysteresis and noise handling	9
Module 4: Timers, PWM and Serial Communication Timing concepts in embedded systems, Software timing and non-blocking programming, Pulse Width Modulation (PWM), LED brightness and actuator control, External interrupts, UART serial communication, Serial Monitor for debugging	6
Module 5: Embedded Applications and Case Studies Principles of embedded application design, Sensor and actuator interfacing, Smart lighting systems, Automatic fan controller, Water level monitoring system, Temperature monitoring and alarm system, Simulation-based embedded system case studies	6

Text Books:

1. Jeremy Blum, Exploring Arduino: Tools and Techniques for Engineering Wizardry, Wiley, 2nd Edition, 2019.

2. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, The AVR Microcontroller and Embedded Systems: Using Assembly and C, Pearson Education, 1st Edition, 2010.

3. Gareth Halfacree, Ben Everard, Get Started with MicroPython on Raspberry Pi Pico, Raspberry Pi Press, 1st Edition, 2021

Reference Books:

1. Steven F. Barrett, Daniel J. Pack, AVR Microcontroller Programming in C, Morgan & Claypool Publishers, 2nd Edition, 2013.

2. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill Education, 2nd Edition, 2016

3. Raspberry Pi Pico C/C++ SDK, Raspberry Pi Foundation, Latest Online Release

4. ATmega328P AVR Microcontroller Datasheet, Microchip Technology Inc., Official device datasheet for the ATmega328P microcontroller used on the Arduino Uno.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Demonstration-based Learning
CD2	Simulation-based Laboratory Sessions (Tinkercad / Wokwi)
CD3	Live Coding and Register-Level Programming
CD4	Guided Hands-on Exercises
CD5	Problem Solving and Debugging Sessions
CD6	Case Studies and Mini Projects
CD7	Quizzes, Assignments and Seminar Presentations

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	3	2	2
CO2	3	2	3	2	3	3	2	2
CO3	3	2	3	3	2	3	1	2
CO4	3	3	2	3	3	2	3	2
CO5	2	3	2	3	3	2	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD7
CO2	CD2, CD3, CD4, CD7
CO3	CD2, CD4, CD5
CO4	CD2, CD3, CD5, CD7
CO5	CD4, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26632

Course Title: Shell Scripting Lab

Pre-requisite(s):

Co- requisite(s): Shell Scripting

Credits: 1.5 L: 0 T: 0 P: 3

Class schedule per week: 3

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of Teacher: Dr. Md Shah Fahad and Dr. S. Kanungo

COURSE OBJECTIVES

This course enables the students to:

1.	Develop familiarity with the Linux shell environment, file system organization, and essential command-line utilities for efficient system navigation and file management.
2.	Apply shell programming constructs, including variables, conditional statements, loops, and command-line arguments, to develop basic shell scripts.
3.	Design modular shell programs using functions, arrays, debugging techniques, and error handling for solving practical computing problems.
4.	Perform file processing and text manipulation using shell utilities, regular expressions, grep, sed, and gawk for data extraction and report generation.
5.	Develop practical shell scripts to automate routine Linux system tasks such as backup, process monitoring, log analysis, scheduling, and network utility operations.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Use Linux shell commands and file system utilities to perform file management and command-line operations efficiently
CO2	Develop shell scripts using decision-making statements, looping constructs, and parameter passing to automate routine computational tasks.
CO3	Construct modular and reusable shell programs using functions, arrays, debugging, and

	exception handling techniques.
CO4	Analyze and process structured and unstructured text files using shell utilities, regular expressions, grep, sed, and gawk for information extraction and reporting.
CO5	Design and implement automation scripts for system administration tasks, including backup, resource monitoring, log analysis, task scheduling, and network utilities in a Linux environment.

SYLLABUS

MODULE	(NO. OF LAB HOURS)
Module 1: Linux Shell Fundamentals and File System Introduction to Linux operating system and shell environment, Understanding different shells (Bash, sh, Korn shell, C shell), Shell prompt and command execution, Linux file system hierarchy, Working with files and directories, File and directory management commands, File viewing and searching commands, Wildcards and pathname expansion, File permissions and ownership, Environment variables, Command history, Input/output redirection, Pipes and filters.	**6**
Module 2: Shell Programming Fundamentals Creating and executing shell scripts, Variables and command substitution, Arithmetic operations, Reading user input, Conditional statements (if, if-else, nested if, elif), Case statement, Looping constructs (for, while, until), Break and continue statements, Command-line arguments, Positional parameters, Option parsing using getopts, Menu-driven shell programs.	**6**
Module 3: Advanced Shell Scripting User-defined functions, Function arguments and return values, Local and global variables, Recursive functions, Arrays, String manipulation, Script debugging using bash -x and set -x, Exit status and error handling, Signal handling using trap, Shell script modularization and sourcing external script libraries.	**6**
Module 4: File Processing and Text Manipulation File test operators, Processing files and directories using shell scripts, Pattern matching with regular expressions, Text processing using **grep**, Stream editing using **sed** (substitution, deletion, insertion, multiline editing), Text processing using **gawk** (fields, records, formatting, report generation), Processing CSV and log files, Search and replace utilities, File comparison and sorting utilities.	**6**

Module 5: Practical Shell Scripting and System Automation	6
Backup and archive automation using tar and gzip, system monitoring scripts (CPU, memory, disk usage, processes), log file analysis and report generation, process management and job control, scheduling tasks using cron, network utility scripts using ping, curl, wget, ssh, and scp, integrated shell scripting mini-projects for Linux system administration and automation.	

Text book:

1. Richard Blum, Christine Bresnahan, "Linux Command Line and Shell Scripting BIBLE", Wiley Publishing, 3rd Edition, 2015.

Reference books:

1.Das, S. *Your UNIX/Linux: The Ultimate Guide*. 3rd Edition, McGraw-Hill Education, 2012. ISBN: 9780077418786

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10

End Semester Exam	% Distribution
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	2	0	2	3	3
CO2	2	2	2	2	0	2	0	2
CO3	3	3	3	3	2	3	1	3
CO4	3	3	3	3	2	3	2	3
CO5	3	3	3	3	2	3	2	3

MAPPING BETWEEN COURSE OUTCOMES AND COURSE DELIVERY METHOD

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD7
CO2	CD2, CD3, CD4, CD7
CO3	CD2, CD4, CD5
CO4	CD2, CD3, CD5, CD7
CO5	CD4, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26634

Course Title: Mobile Application Development using Flutter Lab

Pre-requisite(s): Object-Oriented Programming, DBMS, Web Technologies

Co-requisite(s): Mobile Application Development using Flutter

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Jit Mukherjee and Dr. Indrajit Mukherjee

COURSE OBJECTIVES

This course enables the students to:

1. Develop cross-platform mobile applications using Flutter and Dart
2. Design responsive graphical user interfaces using Fluter Widgets
3. Integrate mobile applications with databases, cloud services, and REST APIs
4. Utilize device hardware features and Firebase services in Flutter applications
5. Design and implement an industry-oriented mobile application

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1 CO2	Develop Flutter applications using Dart programming concepts Design responsive mobile user interfaces using Flutter widgets and navigation.
CO3	Implement database connectivity, REST APIs, and Firebase services.
CO4	Develop secure, optimized Flutter applications that utilize device services.
CO5	Design and deploy a complete mobile application as a mini-project.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module I: Flutter Framework and UI Basics Flutter installation, IDE setup, Dart basics, Widgets and Layouts	6
Module II: Navigation and Data Management Forms, Navigation, Provider, Shared Preferences, SQLite CRUD	8
Module III: Backend Integration Firebase Authentication, Firestore, REST APIs, JSON	8
Module IV: Device Integration Camera, Maps, GPS, Notifications, Secure Storage	8
Module V: Testing and Deployment Testing, APK generation, Deployment, Mini Project	10

Text Books:

1. Eric Windmill, Flutter in Action, Manning Publications, 2020.
2. Carmine Zaccagnino, Programming Flutter: Native, Cross-Platform Apps the Easy Way, Pragmatic Bookshelf, 2020.
3. Marco L. Napoli, Beginning Flutter: A Hands-On Guide to App Development , Wiley, 2019.
4. Fu Cheng, Flutter Recipes, Apress, 2019.

Reference Books:

1. Alessandro Biessek, Flutter Complete Reference. BPB Publication, 2021
2. Priyanka Tyagi, Pragmatic Flutter. CRC Press, 2022
3. Raj Kamal, Mobile Computing, Oxford Press University Press, 2019
4. Bill Phillips, Chris Stewart, and Kristin Marsicano, Android Programming: The Big Nerd Ranch Guide, Pearson Education, 2019

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects

Code	Course Delivery Method
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	2	2	1	1	1
CO2	2	2	3	2	2	1	1	1
CO3	3	3	3	2	2	1	1	1
CO4	3	3	3	3	2	2	2	2
CO5	3	3	3	3	2	2	2	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5,
CO2	CD1, CD2, CD4, CD5, CD7
CO3	CD1, CD2, CD4, CD5, CD7
CO4	CD1, CD2, CD4, CD5, CD7
CO5	CD1, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26636

Course Title: Software Testing and Version Control Laboratory

Pre-requisite(s): Java / Python Programming, Software Design Pattern

Co-requisite(s): Software Testing and Version Control

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. K. Rajnish, Dr. K. S. Patnaik

COURSE OBJECTIVES

This course enables the students to:

1. Use manual testing techniques to find logical errors, edge-case vulnerabilities, and functional gaps in program requirements.
2. To create structural and coverage-driven test suits using control flow metrics and internal code architecture.
3. To manage setup-teardown test cycles and automate unit-level validation using contemporary programmatic testing frameworks.
4. To programmatically target DOM elements and mimic real-world user interactions with web views while acquainting them with cross-browser automation technologies.
5. Establish industry-standard configuration management practices by becoming proficient in branching strategies, remote team collaboration processes, distributed version control systems, and dispute resolution techniques.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Create methodical black-box test scenarios for various application logics by utilizing equivalency classes and boundary conditions.
CO2	Determine full statement and path coverage, map control flow complexity in white-box structure assessments.
CO3	Construct robust unit-level automated test scripts using assertions, test hooks, and mocking setups within native development frameworks.
CO4	Design end-to-end, cross-browser user acceptability web automation processes with testing frameworks and dynamic element locators.
CO5	Organize multi-developer source code synchronization through the use of hosted repository peer reviews, conflict triage strategies, and branching approaches.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Manual Testing Fundamentals Equivalence Class Partitioning (ECP), Boundary Value Analysis (BVA), White-Box Testing & Cyclomatic Complexity	9
Module 2: Automated Unit Testing Introduction to Unit Testing Frameworks, Advanced Unit Testing & Fixtures.	6
Module 3: Web UI Automation Document Object Model (DOM) Traversal & Element Locators, Advanced Locators (XPath) & Dynamic Pages, Simulating User Interactions & Synchronization.	9
Module 4: Local & Advanced Git [distributed version control system] Operations Git Fundamentals & Local Repositories, Git Branching Workflows, Deliberate Conflict Resolution.	9
Module 5: Collaborative Cloud Git & Integrations Remote Repository Synchronization, Team Workflows & Pull Requests.	6

Text Books:

1. Srinivasan Desikan, Gopalaswamy Ramesh, Software Testing: Principles and Practices, Pearson Education, 2008.
2. Scott Chacon and Ben Straub, Pro Git, Pro Gi, Apress, 2014.

Reference Books:

1. Andreas Spillner, Tilo Linz, and Hans Schaefer, Software Testing Foundations: A Study Guide for the Certified Tester Exam, Rocky Nook, 2014
2. Richard E. Silverman, Git Pocket Guide: A Working Introduction, O'Reilly Media, 2013
3. Rex Black, Erik van Veenendaal, and Dorothy Graham, Foundations of Software Testing: ISTQB Certification, Cengage Learning, 2012.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	3	3	3	2	3	1	3
CO2	3	3	3	3	3	3	1	3
CO3	3	3	3	3	3	2	2	3
CO4	3	3	3	3	3	2	2	2
CO5	3	3	3	3	3	2	2	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD4
CO2	CD1, CD4, CD5
CO3	CD1, CD2, CD3, CD4, CD5
CO4	CD1, CD3, CD4, CD5
CO5	CD1, CD2, CD3, CD5

COURSE INFORMATION SHEET

Course Code: CA26638

Course Title: Advanced Programming in Java Lab

Pre-requisite(s): Understanding OOP using Java

Co-requisite(s): Advanced Programming in Java

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Siba Mitra, Dr. J. Bakas

COURSE OBJECTIVES

This course enables the students to:

1.	Develop advanced Java applications using generics, collections, exception handling, and Java 8 features.
2.	Design interactive desktop GUI applications using Swing components and event-driven programming.
3.	Build database-driven applications using JDBC and SQL.
4.	Develop dynamic web applications using Servlets, JSP, and session management.
5.	Design distributed client-server applications using Java networking and integrate concepts through a mini project.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Develop Java applications using generics, collections, lambda expressions, streams, and robust exception handling.
CO2	Design interactive graphical user interface applications using Java Swing components and event handling.
CO3	Implement database applications using JDBC, prepared statements, transactions, and Result Set processing.
CO4	Develop and deploy web applications using Servlets, JSP, cookies, sessions, and MVC architecture.
CO5	Construct distributed client-server applications using Java sockets and networking APIs by integrating multiple concepts into a mini project.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Advanced Java Fundamentals Writing Java programs using generic classes and methods with bounded wildcards. Implement programs demonstrating List, Set, and Map using ArrayList, HashSet, and HashMap. Design a custom exception hierarchy and demonstrate assertions versus exception handling. Implement lambda expressions, functional interfaces, and basic Stream API operations (filter, map, reduce).	6
Module 2: GUI programming Design a Swing-based form using JFrame, JPanel, labels, buttons, and check boxes. Implement menus, toolbars, and actions in a Swing application. Build a Swing application using sliders, spinners, progress bars, and scrollbars. Develop a data-entry application using lists, combo boxes, tables, and trees; add a print option using the Java Print Service API.	9
Module 3: Database Connectivity (JDBC) Establish a JDBC connection and perform CRUD operations using Statement. Implement parameterized queries using PreparedStatement. Execute stored procedures using CallableStatement. Retrieve and update data using ResultSet and ResultSetMetaData.	6
Module 4: Enterprise Web Components (Servlets & JSP) Implement a Servlet demonstrating the complete Servlet life cycle (init, service, destroy). Process HTTP GET/POST requests and generate dynamic responses using Servlets. Implement session tracking using Cookies, Hidden Form Fields, and HttpSession. Developing JSP-based application using scripting tags, directives, and actions.	9
Module 5: Distributed Computing & Network Programming Build a client-server chat application using TCP/IP Sockets. Implement a distributed application using RMI (stubs, skeletons, remote interfaces). Program to fetch and process web content using URL and URLConnection. Program to resolve host information using the InetAddress class.	9

Mini project: Build an end-to-end industry application (e.g., Online Banking Portal, Library Management System, or Distributed Messaging Hub etc.)	

Text Books:

1. Roy, Uttam K. Advanced Java Programming. 1st ed., Oxford University Press, 2015
2. Schildt, Herbert. Java: The Complete Reference. 13th ed., McGraw-Hill Education, 2025

Reference Books:

1. Programming with Java, Beginner to Advanced, Cay S. Horstmann, 7th Edition, Wiley, 2024
2. Java How to Program, Deitel P., Deitel H., Pearson Publications, 2016.
3. Java: One Step Ahead, Seth, Anita, and B. L. Juneja, Oxford University Press, New Delhi, 2017.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	**% Distribution**
Performance	30
Quiz	10
Total	40

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	3	1	1	1	1	2
CO2	1	2	2	3	2	2	2	2
CO3	1	2	2	3	2	3	2	3
CO4	1	2	3	3	2	2	2	3
CO5	1	2	3	3	2	3	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD2, CD3, CD4
CO3	CD1, CD2, CD3, CD4
CO4	CD1, CD2, CD3, CD6
CO5	CD1, CD2, CD3, CD6, CD7

TRACK: MODERN COMPUTING PARADIGM

COURSE INFORMATION SHEET

Course Code: CA26639

Course Title: Deep Learning

Pre-requisite(s): Python Programming, Mathematics for Computing, Machine Learning

Co-requisite(s): Deep Learning Lab

Credits: 03 L: 03 T:0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. P. S. Bishnu, Dr. Poonam Moral

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts, architectures, and principles of deep learning.
2. Explore the application areas of Neural Networks and map a problem.
3. Design Convolution Neural Networks.
4. Design Recurrent Neural Networks.
5. Explore Generative Learning models.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand the foundation and application of deep learning networks
CO2	Design deep neural networks and fit models for specific applications.
CO3	Apply CNN and other pre-trained models.
CO4	Implement RNN based model.
CO5	Apply Generative Learning-based models.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Concepts of Deep Learning Introduction to deep learning, logical computations with neurons, hidden layers, activation functions, loss function and cross entropy loss, perceptron, backpropagation, applications, and use-cases for industry.	4
Module 2: Network Training and Optimization Training a deep neural network (DNN), Gradient descent, Stochastic Gradient descent, Momentum, Adam, Training hyperparameters, Computing derivatives, Back-propagation, Parameter Initialization. vanishing/exploding gradient issues, reusing pretrained layers, optimizers, l1 and l2 regularization, dropout.	10
Module 3: Deep Convolutional Networks and Architectures Convolutional neural networks (CNNs): convolutional layer, filters, stacking, pooling layer, CNN architectures Pretrained Models: VGG16. VGG19. Alex-Net. Res-Net.	10
Module 4: Sequence Learning with Recurrent Neural Networks Recurrent neural networks (RNNs): recurrent neurons, unrolling, input and output sequences, training RNNs, deep RNNs, LSTM cell, GRU cell.	10
Module 5: Deep Generative Learning Representation Learning and Generative Learning: Auto encoders: data representations, linear auto encoder, stacked auto encoders, variational auto encoders.	6

Text Books:

1. C. Bishop, H. Bishop, Deep Learning: Foundations and Concepts, Springer, 1st Edition, 2024.
2. S., J., D. Prince, Understanding Deep Learning, MIT Press, 1st Edition, 2024

Reference Books:

1. Francois Chollet, Deep Learning with Python, Manning Publication, Second Edition, 2021
2. J. Dawani, Hands on Mathematics for Deep Learning, Packt Publishing, 1st Edition, 2020.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards / LCD projectors / OHP projectors
CD2	Assignments / Seminars
CD3	Laboratory experiments / teaching aids
CD4	Industrial / guest lectures
CD5	Industrial visits / in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading:0- No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	3	1	0	1	3
CO2	3	2	3	3	1	0	2	3
CO3	3	2	3	3	1	0	2	3
CO4	3	2	3	3	1	0	2	3
CO5	3	2	3	3	1	0	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD3
CO2	CD3, CD5
CO3	CD3, CD5
CO4	CD2, CD3, CD4
CO5	CD1, CD3, CD5

COURSE INFORMATION SHEET

Course Code: CA26641

Course Title: Large Language Models

Pre-requisite(s): Natural Language Processing

Co-requisite(s): Large Language Model Lab

Credits: 03 L: 03 T: 0 P: 1

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Aditi Panda, Dr. Monu Bhagat

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals and evolution of language models leading to modern Large Language Models.
2. Explain the architecture, training paradigms, and adaptation techniques of transformer-based language models.
3. Illustrate prompt engineering, retrieval-augmented generation, and parameter-efficient adaptation techniques used in LLMs.
4. Develop a basic understanding of frameworks and applications for building LLM-based systems.
5. Analyze the capabilities, limitations, and ethical considerations associated with Large Language Models.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamentals of natural language processing, language modelling and transformer architectures.
CO2	Analyze the pre-training, fine-tuning, and adaptation techniques used in Large Language Models.
CO3	Illustrate prompt engineering, retrieval-augmented generation, and parameter-efficient adaptation techniques used in Large Language Models.
CO4	Explain the development and evaluation of applications using modern Large Language Model frameworks.

CO5	Analyze the ethical, societal, and technical challenges associated with the deployment of Large Language Models.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of Language Models and NLP Introduction to Natural Language Processing (NLP) and its applications, Statistical Language Models and N-gram Models, Word Representations: Word2Vec, GloVe, fastText, Tokenization Strategies: BPE, WordPiece, SentencePiece, Neural Language Models: RNN, LSTM, GRU.	8
Module 2: Transformers and Large Language Models Limitations of Sequential Models and Need for Transformers, Transformer Architecture: Self-Attention and Multi-Head Attention, Positional Encoding, Layer Normalization, and Feed Forward Networks, Encoder-only, Decoder-only, and Encoder-Decoder Models: BERT, GPT, T5.	8
Module 3: Training and Adapting Large Language Models Pre-training Objectives: Masked Language Modeling and Causal Language Modeling, Instruction Tuning and Alignment of LLMs, Prompt Engineering: Zero-shot, Few-shot, and Chain-of-Thought Prompting, Parameter-Efficient Adaptation: Prompt Tuning, Prefix Tuning, and LoRA.	8
Module 4: LLM Applications and Development Hugging Face Ecosystem and LLM Inference, Retrieval-Augmented Generation (RAG): Concepts and Pipeline, Knowledge-Augmented Question Answering, Applications of LLMs: Chatbots, Summarization, Translation, and Code Generation.	8
Module 5: Advanced Large Language Models and Responsible AI Overview of Recent LLMs: GPT, Llama, Claude, Gemini, and Mistral, Evaluation of Large Language Models, Hallucination, Interpretability, and Limitations of LLMs, Ethical AI: Bias, Safety, Privacy, and Responsible Deployment.	8

Text Books:

1. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, 2022.
2. Delip Rao and Brian McMahan, Natural Language Processing with PyTorch, O'Reilly Media, 2019.
3. Yoav Goldberg, Neural Network Methods for Natural Language Processing, Morgan & Claypool Publishers, 2017.

Reference Books:

1. Jacob Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.
2. Jurafsky, D. and Martin, J. H., Speech and Language Processing, 3rd Edition (Draft), Pearson, 2000
3. Denis Rothman, Transformers for Natural Language Processing, 2nd Edition, Packt Publishing, 2021.
4. Sebastian Raschka, Build a Large Language Model (From Scratch), Manning Publications, 2024.
5. Chip Huyen, AI Engineering: Building Applications with Foundation Models, O'Reilly Media, 2025.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Code	Course Delivery Method
CD1	Classroom Lectures
CD2	Interactive Tutorials and Discussions
CD3	Case Studies on Large Language Model Applications
CD4	Demonstration of LLM Frameworks and Tools (e.g., Hugging Face)
CD5	Hands-on Programming Assignments
CD6	Student Seminars and Presentations
CD7	Self-learning through Research Papers and Technical Writing

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	3	3	2	2
CO2	3	2	3	2	2	3	1	2
CO3	3	3	3	2	3	3	3	2
CO4	2	3	3	3	2	2	2	2
CO5	3	2	2	2	3	3	2	1

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2
CO2	CD1, CD2, CD4
CO3	CD1, CD2, CD4, CD5
CO4	CD3, CD4, CD5, CD6
CO5	CD1, CD2, CD3, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26643

Course Title: Generative AI

Pre-requisite(s): Python Programming

Co-requisite(s): Machine Learning

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: CSE

Name of the Teachers: Dr. C. Lavania, Dr. Monu Bhagat

COURSE OBJECTIVES

This course enables the students to:

1. Understand the concept of Generative AI and gain an understanding of the type of tasks that can be performed with Generative AI
2. Gain an understanding of the core building blocks of modern Generative AI systems
3. Understand the different types of modalities that are commonly used with Generative AI
4. Understand the limitations, and inculcate the responsible use of Generative AI
5. Gain the fundamental understanding to build a Generative AI system

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Identify problems that can be solved with Generative AI and build basic generative models
CO2	Build and train LLMs
CO3	Create transformers and diffusion-based models for NLP and computer vision tasks
CO4	Build prompts for modern Generative AI based chat bots and use Generative AI for text/visuals generation
CO5	Build responsible and explainable Generative AI systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to GenAI What is Generative AI? discriminative model, generative model (introduction to VAE, GAN), generative model vs Generative AI, examples and applications of Generative AI (such as chat bots, image/text generation), introduction to sequence prediction and autoregressive model.	8
Module 2: Transformers and Diffusion What are transformers? Introduction to the transformer architecture, positional encoding, types of attention (including multi-headed attention, self-attention, cross-attention and masked attention), concept of multimodality, multimodal transformer architectures, introduction to the concept of foundation models, introduction to BERT, GPT3, Vision Transformer (ViT). Diffusion process, use of diffusion to generate audio, videos, images (including the differences in the process for each modality), latent diffusion models, conditional diffusion models and their applications (text-to-image generation, image-to-image, audio-visual generation).	10
Module 3: LLMs and Training LLMs What is a language model (LM)? classical use cases of language models (such as ASR), introduction to Hidden Markov Models (HMM)? What constitutes a large language model (LLM), usage of deep neural networks for LLM, use of LLMs in Generative AI, concept of tokenization and vocabulary. Supervised fine tuning, introduction to reinforcement learning from human feedback (RLHF) for training LLMs.	6
Module 4: Introduction to prompts System instructions, user instructions, embedding examples in a prompt. Example prompts for popular LLMs (claude, GPT) Modern Usage**:** usage in image generation (diffusion and GAN based), text generation, audio-generation, audio-visual generation, audio/audio-visual/text translation (such as Sora, MuAViC), encoder only models, decoder only models, Llama, Mistral, Vall-E, multimodal generative models GPT5, multimodal models - Gen2 (vision-language), AV-CDiT (audio-visual), VATT (audio-visual-text), multimodal generative models GPT5.	8
Module 5: Limitations, Issues and Responsible Generative AI	8

Non-Deterministic nature-LLMs, diffusion models, VAEs and GANs. Requirement of guardrails, Deepfake-audio-video-images, face swap, malicious usage of biometric content, voice cloning. Explainability, watermarking, ethical implications, impact on creativity, bias reduction, accountability, privacy and security.	

Text Books:

1. William Brown, Generative AI Handbook, 2024
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

Reference Books:

1. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose and Play, O'Reilly, Second Edition, 2019.
2. Sebastian Raschka, Build a Large Language Model (From Scratch), Manning Pubns Co, 2024.
3. Paul Iusztin and Maxime Labonne, LLM Engineer's Handbook: Master the art of engineering large language models from concept to production, Packt Publishing Limited, 2024.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

<u>Direct Assessment</u>

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	20
Assignment(s)	10
Seminar before a committee	20
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

<u>Indirect Assessment</u>

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini Projects
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulations

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	2	2	2	3
CO2	3	3	3	3	2	2	2	3
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	3	2	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6
CO2	CD1, CD2, CD4, CD6
CO3	CD1, CD2, CD4, CD6
CO4	CD1, CD2, CD4, CD6
CO5	CD1, CD2, CD3, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26645

Course Title: Computer Vision

Pre-requisite(s): Image Processing, Machine Learning

Co-requisite(s): Computer Vision Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. K. K. Senapati, Dr. Shamama Anwar

COURSE OBJECTIVES

This course enables the students to:

1. Understand the Fundamental Concepts of Computer Vision.
2. Apply the concept of color and early vision through case studies.
3. Analyze and extract features from images using computer vision algorithms.
4. Design and Develop methods for Computer Vision practical problems.
5. Evaluate the performance of computer vision systems and optimize.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand Computer Vision fundamentals.
CO2	Apply vision enhancement techniques
CO3	Implement computer vision models
CO4	Analyze Vision Models
CO5	Design solutions for real world problems

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Computer Vision Image Formation and Image Models-Cameras, Geometric Camera Models, Geometric Camera Calibration, Measuring Light, Sources, Shadows and Shading, AI based Case Studies	8
Module 2: Color Physics of Color, Human Color Perception, Representing Color, Color Model, Early Vision: Linear filters, Edge Detection, Use of AI through Case Studies	8
Module 3: Multiple Images Geometry of Multiple Views, Affine Structure from Motion, Projective Structure from Motion, AI based Applications/Case Studies	8
Module 4: Mid-Level Vision Segmentation by Clustering, Segmentation by fitting a model, Segmentation and fitting using probabilistic methods, AI Model based Case Studies	8
Module 5: High Level Vision Model Based Vision, Smooth Surfaces and their outlines, Use of AI in Vision, ML and Vision, Case Studies	8

Textbooks:

1. D A Forsyth, Jean Ponce, Computer Vision, Person Education, Second Edition, 2025

Reference Books:

2. Simon J D Pence, Computer Vision, Cambridge University Press, 2012

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30

Continuous Internal Assessment	% Distribution
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	1	0	0	0
CO2	3	3	1	2	2	0	0	0
CO3	2	3	2	2	3	1	0	0
CO4	2	3	2	3	3	1	0	0
CO5	2	3	3	2	3	1	1	0

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD5, CD6, CD7
CO2	CD1, CD2, CD5, CD6, CD7

Course Outcomes	Course Delivery Method
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6
CO5	CD1, CD2, CD3, CD4, CD5, CD6

COURSE INFORMATION SHEET

Course Code: CA26640

Course Title: Deep Learning Lab

Pre-requisite(s): Python programming

Co-requisite(s): Deep Learning

Credits: 1.5 L:0 T:0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Shamama Anwar and Dr. Partha Sarthi Bishnu

COURSE OBJECTIVES

This course enables the students to:

1. Develop deep learning models using Python, TensorFlow, and Keras.
2. Implement computer vision and natural language processing applications using deep learning.
3. Apply autoencoders for feature learning and dimensionality reduction.
4. Evaluate and optimize deep learning models using modern training techniques.
5. Design practical deep learning solutions for real world applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Implement deep learning models using Python, TensorFlow, and Keras.
CO2	Develop CNN and RNN/LSTM models for vision and text applications.
CO3	Apply autoencoders for representation learning and feature extraction.
CO4	Evaluate and optimize deep learning models using appropriate techniques.
CO5	Design and implement deep learning solutions for real-world problems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Deep Learning Environment and Feedforward Networks 1. Installation and setup of Python, TensorFlow, Keras, Jupyter Notebook. 2. Implement data preprocessing, feature engineering, and dataset splitting (training, validation, and testing) using Scikit-learn. 3. Implement a feedforward neural network and study the effects of overfitting, dropout, early stopping, and weight regularization.	8
Module 2: Convolutional Neural Networks and Transfer Learning 1. Implement a convolutional neural network for image classification. 2. Apply image preprocessing and data augmentation techniques using Keras ImageDataGenerator. 3. Perform transfer learning using a pretrained CNN (VGG16/ResNet50/MobileNet) and fine-tune the model.	6
Module 3: Sequence Modeling with Recurrent Neural Networks 1. Perform text preprocessing, tokenization, one-hot encoding, and word embedding generation. 2. Implement an RNN/LSTM model for sentiment analysis or text classification. 3. Compare the performance of LSTM and GRU models on a sequence prediction or text classification dataset.	6
Module 4: Deep Representation Learning with Autoencoders 1. Implement a basic Autoencoder for dimensionality reduction and feature learning. 2. Develop a Denoising Autoencoder for image denoising. 3. Implement a Convolutional Autoencoder for image reconstruction and latent feature extraction.	6
Module 5: Deep Learning Model Optimization and Deployment 1. Compare different optimizers (SGD, RMSprop, Adam) and learning rate schedules for model training. 2. Implement transfer learning, model checkpointing, model serialization, and evaluate ensemble learning techniques.	4

Text Books:

1. François Chollet, Deep Learning with Python, 2nd Edition, Manning Publications, 2021.

2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition, O'Reilly Media, 2022.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.

2. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, 2018.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading:0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	3	1	1	1	2
CO2	2	3	3	3	2	1	1	2
CO3	2	3	3	3	1	1	1	2
CO4	2	3	2	3	1	1	1	3
CO5	2	3	3	3	2	1	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD5, CD6
CO2	CD1, CD2, CD5
CO3	CD2, CD5, CD6
CO4	CD2, CD3, CD5
CO5	CD3, CD4, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26642

Course Title: Large Language Models Lab

Pre-requisite(s): Python Programming

Co-requisite(s): Large Language Model

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III/ VI

Branch: MCA

Name of the Teachers: Dr Komal Naaz, Dr Indrajit Mukherjee

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts of Large Language Models (LLMs), including corpus preprocessing, tokenization techniques, and embedding methods.
2. Develop an understanding of text representation techniques and embedding models to perform semantic similarity, semantic search, and document retrieval.
3. Enable students to apply prompt engineering techniques and utilize pre-trained transformer models for effective text generation.
4. Provide hands-on experience in implementing transformer-based models for text summarization and question answering tasks.
5. Enable students with practical knowledge of fine-tuning transformer models, performing sentiment analysis, and developing Retrieval-Augmented Generation (RAG) applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Implement text preprocessing, tokenization, and corpus preparation techniques using open-source LLM development tools.
CO2	Apply embedding models and semantic similarity techniques to build semantic search and document retrieval systems.
CO3	Utilize prompt engineering strategies and transformer-based language models to generate coherent and contextually relevant text.
CO4	Develop applications for text summarization and question answering using pre-trained transformer models.
CO5	Fine-tune transformer models for sentiment analysis and construct Retrieval-Augmented Generation (RAG) applications using open-source frameworks.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Preprocessing Corpus Familiarity with LLM development environment (Installation of Python libraries and Hugging Face ecosystem) Text cleaning, tokenization (WordPiece, BPE, SentencePiece, Compare vocabulary sizes), stemming, normalization, stop-word removal, Information retrieval using regular expression.	6
Module 2: Embeddings TF-IDF, Word2Vec, GloVe, Sentence Embedding, Cosine Similarity, Semantic Search, Document Retrieval: Build Embedding Index, Retrieve top-k documents	6
Module 3: Prompt Engineering and Text Generation Prompt Engineering Fundamentals: Zero-Shot, One-Shot, Few-Shot; Chain-of-thought, Self-Consistency, Role-prompting, Structured Prompting Generate text using GPT Models, Introduction to Temperature, Top-K, Top-P, Max Tokens	12
Module 4: Summarization and Question Answering Using Models T5, FLAN-T5, BART	6
Module 5: Fine-tuning and RAG Fine-tune BERT for sentiment analysis (Dataset: IMDb or SST-2), Parameter Efficient Fine-tuning (PEFT): LoRA, QLoRA) Semantic Search and Retrieval-Augmented Generation	12

Text Books:

1. J. Alammar, M. Grootendorst, *Hands-on large language models: Language understanding and generation*. O'Reilly Media, 2024.

Reference Books:

1. (Online) Huggingface LLM Tutorial: https://huggingface.co/learn/llm-course/chapter1/1
2. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, 2022.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	2	1	0	2	3
CO2	3	1	1	2	0	0	1	3
CO3	3	3	3	3	3	2	2	3
CO4	3	3	3	3	3	2	2	3
CO5	3	3	3	3	3	2	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5
CO2	CD1, CD2, CD3, CD5
CO3	CD1, CD2, CD3, CD5, CD4, CD6
CO4	CD1, CD2, CD3, CD5, CD4, CD6, CD7
CO5	CD1, CD2, CD3, CD5, CD$, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26644

Course Title: Generative AI Lab

Pre-requisite(s): Machine Learning

Co-requisite(s): Generative AI

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. C. Lavania, Dr. S. K. Sahana

COURSE OBJECTIVES

This course enables the students to:

1. Understand the type of tasks that can be performed with Generative AI.
2. Understand the core building blocks of Generative AI systems.
3. Gain an understanding of the different modalities that are commonly used with Generative AI.
4. Understand the limitations and encourage responsible use of Generative AI.
5. Gain the understanding to build a Generative AI system.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Identify the problems that can use Generative AI and build classical models for generation.
CO2	Build and train transformers and diffusion-based models.
CO3	Build and train language models.
CO4	Build prompts for modern Generative AI based chatbots and generate text/visuals using Generative AI.
CO5	Build responsible Generative AI systems.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introductory Formulations: VAE, GAN (such as for generating animal images), Sequence Prediction, Autoregressive Model (such as for forecasting temperature). Example Exercises: 1) Implement a basic VAE using fully connected layers to compress and reconstruct data while learning a continuous latent space. 2) Implement a VAE with convolutional layers in the encoder and transposed convolutions in the decoder, making it ideal for spatial grid data like images. 3) Implement a GAN that uses spatial convolutions and spatial stride transpositions for crisp image generation. 4) Implement an LSTM/GRU based model for sequence-to-sequence or sequence-to-one tasks (such as temperature forecasting). 5) Implement an autoregressive sequential architecture that shifts inputs or applies causal padding so that 1D filters can never see into the future.	6
Module 2: Transformers Classical Transformers, Positional Encoding, attention (multi-headed attention, self-attention, cross-attention and masked attention), Tokenization, BERT, VisionTransformer (ViT), multimodal transformers. **Diffusion:** Latent diffusion model, text-to-image generation (such as generating image from description), image-to-image generation (such as generating image from the information available in another image) Example Exercises: 1) Implement a classical transformer with single headed attention for generating a sentence from a given input sentence. a. Try different positional encoding (such as sinusoidal, rotary) and observe the difference. b. Implement different tokenizers (such as BPE, WordPiece for text) and analyze their impact in performance. 2) Use a video as input and build a transformer architecture with cross-attention to extract multimodal features for a downstream scene classification task. 3) Finetune BERT/ViT for a given text/image dataset.	10

4) Build a diffusion model to generate images with no external input. 5) Build a diffusion model that takes a prompt (text/image) as input and produces the corresponding image	
Module 3: Language Models Hidden Markov Models (HMM), Small Language Model (SLM), Large Language Model (LLM), Vocabulary Creation, LLM Training (supervised fine tuning, RLHF). Example use case – Train an LLM to generate text based on an input prompt Example Exercises: 1) Build an HMM for next word prediction 2) Finetune a SLM for text classification 3) Build an LLM for text classification from scratch 4) Build an SLM from an LLM	6
Module 4: Prompt Engineering and Modern Usage System instructions, user instructions, embedding examples in a prompt. Example prompts for popular LLMs (claude, GPT). Decoder only model (such as GPT for text generation), introduction to Llama, Mistral, multimodal generative models GPT5, multimodal models - Gen2 (vision-language), AV-CDiT (audio-visual), introduction to langchain. Example Exercises: 1) Create a prompt for parsing specific information from a text input (such as extracting all named entities) 2) Create a prompt for creating a ranking between objects (given as input). 3) Use Llama for image generation 4) Use langchain to combine a text based LLM and a Vision based LLM for caption generation.	8
Module 5: Limitations, issues and responsible usage Non-Determinism in LLMs, VAE, GAN, Diffusion models (show how multiple inference runs through the same model can result in different responses), Deepfake identification (such as identifying fake images), Watermarking, Explainability.ss Example Exercises: 1) Build a model to identify fake images 2) Add watermarks to generated images 3) Use a modern LLM (such as Llama, claude, GPT) to produce different outputs from the same input thereby demonstrating the non-	9

deterministic nature of these models. Analyze the cause of non-determinism. What steps can be taken to reduce it in the experiments?	

Text Books:

1. William Brown, Generative AI: A Roadmap for Learning Resources, 2024
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016
3. Bert Gollnick, Generative AI with Python: The Developer's Guide, Rheinwerk Computing, 2025

Reference Books:

1. Mark Liu, Learn Generative AI with PyTorch, Manning Pubns Co, 2024
2. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose and Play, O'Reilly, Second Edition, 2019
3. Sebastian Raschka, Build a Large Language Model (From Scratch), Manning Pubns Co, 2024
4. Paul Iusztin and Maxime Labonne, LLM Engineer's Handbook: Master the art of engineering large language models from concept to production, Packt Publishing Limited, 2024
5. https://docs.pytorch.org/tutorials/beginner/basics/intro.html
6. https://www.tensorflow.org/tutorials/quickstart/beginner

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	2	2	2	3
CO2	3	3	3	3	2	2	2	3
CO3	3	3	3	3	2	2	2	3
CO4	3	3	3	3	2	2	2	3
CO5	3	3	3	3	3	2	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD5, CD6
CO2	CD1, CD2, CD3, CD5, CD6, CD7

Course Outcomes	**Course Delivery Method**
CO3	CD1, CD2, CD3, CD5, CD6, CD7
CO4	CD1, CD2, CD5, CD6, CD7
CO5	CD1, CD2, CD3, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA6646

Course Title: Computer Vision LAB

Pre-requisite(s): Python Programming

Co-requisite(s): Computer Vision

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. K. K. Senapati, Dr. Shamama Anwar

COURSE OBJECTIVES

This course enables the students to:

1.	Implement Computer Vision Tools
2.	Developing Skills in Computer Vision Applications
3.	Apply Vison techniques /programming in real world applications
4.	Learn vision stacks or frameworks
5.	Design Model/Projects using vision libraries/skills

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Design Computer Vision Models
CO2	Develop programs for Computer Vision applications
CO3	Demonstrate Computer vision projects
CO4	Detect bugs/errors in computer vision problems
CO5	Deploy computer vision libraries/stacks

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction and Environment Setting Installation and Introduction Programs of a CV tool	9

Module 2: Image Processing using Color Models Programs/Projects based on Color Model and Single Image	6
Module 3: Motion Analysis and Multi-Image Processing Programs/ Projects based on Motion and Multiple Images	9
Module 4: Image Segmentation and Probabilistic Modeling Programs/Projects based on Segmentation and Probabilistic Models	6
Module 5: AI-Based Computer Vision Applications Programs/ Projects based on Computer Vision with AI components	9

Textbooks:

1. Computer Vision: Algorithms and Applications by Richard Szeliski , Second Edition, Springer, 2021

Reference Books:

1. Deep Learning, by Goodfellow, Bengio, and Courville., 2018
2. Dictionary of Computer Vision and Image Processing, by Fisher et al., 2015

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	**% Distribution**
Performance	30
Quiz	10

End Semester Exam	% Distribution
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	1	0	2	3
CO2	3	3	3	3	1	0	3	3
CO3	1	1	2	3	3	2	3	2
CO4	2	3	2	3	0	0	2	2
CO5	1	2	3	3	0	0	1	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6, CD7

Course Outcomes	Course Delivery Method
CO2	CD1, CD2, CD5, CD6
CO3	CD3, CD4, CD5, CD6
CO4	CD2, CD5, CD6, CD7
CO5	CD4, CD5, CD6, CD7

TRACK: Web Application

COURSE INFORMATION SHEET

Course Code: CA26647

Course Title: Introduction to Django

Pre-requisite(s): Python Programming, Frontend Design 2

Co-requisite(s): Full Stack using Django Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Sandip Ghosal, Dr. Rathindranath Dutta

COURSE OBJECTIVES

This course enables the students to:

1. Understand fundamentals of web application development and the Django framework.
2. Develop an understanding of Django ORM for database design and data management.
3. Create dynamic web applications using views, templates, and forms.
4. Familiarize with user authentication, session management, and other built-in Django features.
5. Apply REST APIs, security mechanisms, deployment concepts, and best practices for Django application development.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Understand Django architecture and web application development.
CO2	Design and manage databases using Django ORM.
CO3	Develop dynamic web applications using views, templates, and forms.
CO4	Implement authentication and session management.
CO5	Develop basic REST APIs and understand security and deployment concepts.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Django Web application fundamentals, client-server architecture, static vs dynamic websites, Python overview, Django features, MVT architecture, installation, virtual environment, project and app creation, project structure, development server, URL routing.	8
Module 2: Models and Database Management Introduction to databases, Django ORM, models, field types, model relationships, migrations, Django Admin, CRUD operations, QuerySets, basic queries, model validation.	8
Module 3: Views, Templates and Forms Function-based and class-based views, request-response cycle, URL dispatcher, templates, Django Template Language, template inheritance, static and media files, Django forms, model forms, form validation.	8
Module 4: Authentication and Advanced Features User authentication, registration, login/logout, permissions, groups, sessions, cookies, messages framework, generic views, pagination, middleware, signals, file upload.	8
Module 5: REST APIs, Security and Deployment Django security (CSRF, SQL injection, XSS), error handling, logging, introduction to Django REST Framework, REST APIs, JSON responses, deployment concepts, production environment, and best practices.	8

Text Books:

1. Adrian Holovaty and Jacob Kaplan-Moss, The Definitive Guide to Django: Web Development Done Right, 2nd Edition, Apress, 2009.
2. William S. Vincent, Django for Beginners: Build Websites with Python and Django, 4th Edition, WelcomeToCode, 2020.
3. Antonio Melé, Django 5 By Example, 5th Edition, Packt Publishing, 2024.

Reference Books:

1. Nigel George, Mastering Django: Core, Packt Publishing, 2024.
2. Harry J. W. Percival, Test-Driven Development with Python: Obey the Testing Goat, 2nd Edition, O'Reilly Media, 2017.

3. Daniel Roy Greenfeld and Audrey Roy Greenfeld, Two Scoops of Django 3.x: Best Practices for the Django Web Framework, Two Scoops Press, 2021.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Classroom Lectures
CD2	Interactive Discussions and Question–Answer Sessions
CD3	Demonstration of Django Concepts and Case Studies
CD4	Laboratory Demonstrations and Hands-on Practice
CD5	Tutorials and Problem-Solving Sessions
CD6	Assignments, Quizzes, and Self-Learning Activities
CD7	Seminar Presentations and Project-Based Learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	0	1
CO2	3	3	2	3	0	0	0	1
CO3	3	3	3	3	2	0	1	1
CO4	2	2	3	3	1	0	3	1
CO5	2	2	3	3	2	1	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3
CO2	CD1, CD3, CD4, CD5
CO3	CD1, CD3, CD4, CD5, CD6
CO4	CD1, CD3, CD4, CD6
CO5	CD1, CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26649

Course Title: Introduction to Laravel

Pre-requisite(s): Understanding OOP using Java

Co-requisite(s): Full Stack using Laravel Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Rathindranath Dutta, Dr. Sandip Ghosal

COURSE OBJECTIVES

This course enables the students to:

1. Learn PHP language fundamentals and Object-Oriented Programming (OOP) concepts.
2. Understand web application development and the Laravel MVC framework architecture.
3. Develop core proficiency in database design, migrations, and data management using Eloquent ORM.
4. Build robust and secure dynamic web interfaces using controllers, Blade templates, and form validation.
5. Apply user authentication, state management, RESTful API design, and modern deployment practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Demonstrate proficiency in PHP programming and Laravel architecture.
CO2	Design and manage databases using Eloquent ORM.
CO3	Develop dynamic web applications using controllers, Blade UI, and forms.
CO4	Implement authentication and session management.
CO5	Develop basic REST APIs and understand security and deployment concepts.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: PHP Fundamentals and Introduction to Laravel PHP basics: Syntax, variables, data types, operators, control structures (conditionals, loops), arrays (indexed, associative), built-in and user-defined functions. Object-Oriented PHP: Classes, objects, properties, methods, inheritance, access modifiers, constructor/destructor, namespaces, and traits. Laravel Overview: Web application fundamentals, client-server architecture, MVC pattern, Composer, directory structure, Artisan CLI, and environment configuration (env).	8
Module 2: Models and Database Management Introduction to databases; Eloquent ORM principles; creating models and defining field attributes; database migrations (create/modify tables); schema builder; database seeding; model factories for testing; relationship mapping (one-to-one, one-to-many, many-to-many); Fluent Query Builder; basic CRUD operations; model events; and query scopes.	8
Module 3: Views, Templates and Forms Request–response lifecycle; URL dispatching/routing; basic route parameters; route groups and route naming; Controllers: Basic controllers, resource controllers, and dependency injection; Blade template fundamentals; template inheritance; layouts; control structures; components; Form handling; Form Request validation; and error display handling.	8
Module 4: Authentication and Advanced Features User authentication, registration, login/logout, permissions, sessions, cookies, flash messages framework, authorization, role-based access layout, pagination, middleware, file storage.	8
Module 5: REST APIs, Security and Deployment Web application security: CSRF protection, SQL injection prevention, and XSS prevention. Error handling and logging in web applications. REST APIs in Laravel: API routing, JSON data exchange, API Resources for response transformation, and token-based authentication using Laravel Sanctum. Deployment concepts: Production setup, performance optimization using cache commands (config, routes, views), and deployment best practices.	8

Text Books:

1. Matt Stauffer, Laravel: Up & Running: A Framework for Building Modern PHP Apps, O’Reilly Media, 2nd Edition, 2019

2. Luke Welling and Laura Thomson, PHP and MySQL Web Development, Addison-Wesley Professional, 5th Edition, 2016

3. Jesse Griffin, Laravel: Build Modern PHP Applications, Packt Publishing, 2024.

Reference Books:

1. Matt Zandstra, PHP Objects, Patterns, and Practice, 5th Edition, Apress, 2016.

2. Mark Masse, REST API Design Rulebook, O'Reilly Media, 2011.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Classroom Lectures
CD2	Interactive Discussions and Question–Answer Sessions
CD3	Demonstration of Django Concepts and Case Studies
CD4	Laboratory Demonstrations and Hands-on Practice
CD5	Tutorials and Problem-Solving Sessions
CD6	Assignments, Quizzes, and Self-Learning Activities
CD7	Seminar Presentations and Project-Based Learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	0	1
CO2	3	3	2	3	0	0	0	1
CO3	3	3	3	3	2	0	1	1
CO4	2	2	3	3	1	0	3	1
CO5	2	2	3	3	2	1	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD4
CO2	CD1, CD3, CD4, CD5
CO3	CD1, CD3, CD4, CD5, CD6
CO4	CD1, CD3, CD4, CD6
CO5	CD1, CD2, CD3, CD4, CD7

COURSE INFORMATION SHEET

Course Code: CA26651

Course Title: Frontend Engineering with React JS

Pre-requisite(s): Programming Fundamentals

Co-requisite(s): Full Stack using React JS Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Lopamudra Hota, Dr. Subrajeet Mohapatra

COURSE OBJECTIVES

This course enables the students to:

1. Understand the principles of modern frontend engineering and React's component-based architecture.
2. Develop reusable and interactive user interfaces using JSX, functional components, props, state, and React Hooks.
3. Build scalable Single Page Applications (SPAs) through routing, asynchronous programming, and REST API integration.
4. Apply advanced React concepts including state management, performance optimization, and testing.
5. Design, optimize, and deploy production ready React applications following modern frontend engineering best practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain front-end engineering concepts, React architecture, JSX, and component-based development.
CO2	Develop reusable React components using Hooks, state, props, and event handling.
CO3	Build interactive Single Page Applications with routing, RESTful API integration, and asynchronous data handling.
CO4	Implement scalable applications using Context API, Redux Toolkit, authentication, testing, and performance optimization.

CO5	Design, optimize, test, and deploy production ready React applications following modern frontend engineering practices.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Foundations of React Development Evolution of Frontend Engineering, Introduction to React, Virtual DOM and React Architecture, Node.js and npm, React Development Environment, Vite Project Setup, JSX Syntax and Rendering Elements	8
Module 2: React Components, State, and Hooks React Rendering Process, Component State, Event Handling, Controlled Components, Forms in React, Lists and Keys, React Hooks, useState, useEffect, Side Effects, Custom Hooks, Hook Best Practices	8
Module 3: React Routing and API Integration Thinking in React, Component Hierarchy, React Router DOM, Nested Routes, REST APIs, Fetch API, Asynchronous Programming, Loading States, Error Handling, CRUD Application	8
Module 4: Advanced React State Management and Performance Reducers, useReducer Hook, Managing Global State, Virtual DOM Reconciliation, Memoization (React.memo, useMemo, useCallback), Lazy Loading and Code Splitting, JWT Overview	8
Module 5: React Styling, Testing, and Deployment Styling React Applications, Tailwind CSS, Responsive UI Design, Accessibility (ARIA and WCAG), Unit Testing using Jest, React Testing Library, Deployment using Vercel, Netlify, GitHub Pages	8

Text Books:

1. Alex Banks, Eve Porcello, Learning React, O'Reilly Media, 2024.

Reference Books:

1. Stoyan Stefanov, React Up & Running, O'Reilly Media, 2021

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments/Seminars
CD3	Laboratory experiments/teaching aids
CD4	Industrial/guest lectures
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Project-based learning

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	2	0	0	0	2
CO2	3	2	3	3	2	0	0	2
CO3	2	3	3	3	2	1	0	2
CO4	2	3	3	3	2	1	1	3
CO5	2	2	3	3	3	2	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD5, CD7

COURSE INFORMATION SHEET

Course Code: CA26648

Course Title: Full Stack using Django Lab

Pre-requisite(s): Python Programming, Front End Design

Co-requisite(s): Introduction to Django

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Sandip Ghosal, Dr. Rathindranath Dutta

COURSE OBJECTIVES

This course enables the students to:

1. Familiarize with the Django development environment and project structure.
2. Provide hands-on experience in database design and ORM-based application development.
3. Build dynamic web interfaces using templates and forms.
4. Develop practical skills in implementing authentication and advanced Django features.
5. Build REST APIs and develop an end-to-end Django web application following best practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Configure the Django development environment and create basic web applications.
CO2	Design database models and perform CRUD operations using Django ORM.
CO3	Develop dynamic web pages using templates and forms with validation.
CO4	Implement user authentication, session management, and advanced Django features.
CO5	Develop REST APIs and integrate Django components to build and deploy a functional web application.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Django Environment and Basic Application Development Installation of Python and Django, virtual environment, creating Django projects and applications, project structure, development server, URL routing, creating basic views and templates.	9
Module 2: Models and Database Operations Creating models, configuring databases, migrations, Django Admin, CRUD operations using ORM, QuerySets, model relationships, basic database queries.	6
Module 3: Templates and Forms Template inheritance, static and media files, Django Template Language (DTL), creating forms, model forms, form validation, handling user input and displaying dynamic data.	9
Module 4: Authentication and Advanced Features User registration, login/logout, authentication, permissions, sessions, cookies, messages framework, file upload, pagination, generic views.	6
Module 5: REST APIs and Mini Project Creating REST APIs using Django REST Framework, JSON responses, API testing, implementing security features, deploying a Django application (basic), development of a mini web application integrating learned concepts.	9

Text Books:

1. Adrian Holovaty and Jacob Kaplan-Moss, The Definitive Guide to Django: Web Development Done Right, 2nd Edition, Apress, 2009.
2. William S. Vincent, The Definitive Guide to Django: Web Development Done Right, 2nd Edition, Apress, 2009.

Reference Books:

1. Nigel George, Mastering Django: Core, Packt Publishing, 2024.
2. Harry J. W. Percival, Test-Driven Development with Python: Obey the Testing Goat, 2nd Edition, O'Reilly Media, 2017.
3. Daniel Roy Greenfeld and Audrey Roy Greenfeld, Two Scoops of Django 3.x: Best Practices for the Django Web Framework, Two Scoops Press, 2021.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	3	0	0	0	1
CO2	3	3	3	3	1	0	0	1
CO3	3	3	3	3	2	0	1	1
CO4	2	2	3	3	2	0	3	1
CO5	2	3	3	3	3	2	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD5, CD6, CD7
CO2	CD1, CD2, CD4, CD5
CO3	CD1, CD2, CD3, CD4, CD5
CO4	CD1, CD2, CD3, CD4, CD6
CO5	CD2, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26650

Course Title: Full Stack using Laravel Lab

Pre-requisite(s): Python in Linux

Co-requisite(s): Introduction to Laravel

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Rathindranath Dutta, Dr. Sandip Ghosal

COURSE OBJECTIVES

This course enables the students to:

1. Develop basic webpages with PHP.
2. Build Laravel web applications by implementing the MVC architecture, routing, controllers, Blade templates, and form handling.
3. Design and manage relational databases using Eloquent ORM.
4. Implement secure and scalable web applications with authentication, authorization, middleware, sessions, file storage, and validation.
5. Develop, test, and deploy RESTful APIs using Laravel Sanctum while applying web security measures, error handling, logging, and deployment best practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Develop PHP webpages using procedural and object-oriented programming concepts.
CO2	Create Laravel applications using MVC architecture and Artisan tools.
CO3	Design and manage relational databases using Eloquent ORM, migrations, and relationships.
CO4	Develop secure web applications with authentication, authorization, form validation, and file management.
CO5	Build and deploy RESTful APIs using Laravel Sanctum while applying web security and deployment best practices.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: PHP Fundamentals and Laravel Setup Write PHP programs demonstrating variables, data types, operators, and type conversion. Develop programs using conditional statements and different looping constructs. Implement programs using indexed arrays, associative arrays, and multidimensional arrays. Create user-defined functions for mathematical and string processing operations. Develop Object-Oriented PHP programs demonstrating classes, objects, constructors, inheritance, access modifiers, namespaces, and traits. Install Composer and Laravel framework, configure the development environment, and explore Laravel project directory structure. Use Artisan CLI commands to create and manage a Laravel project.	6
Module 2: Models and Database Operations Configure database connectivity in Laravel using the .env file. Create database tables using Laravel Migrations. Modify existing tables using migration rollback and alteration commands. Develop Eloquent Models and perform CRUD operations. Populate databases using Seeders and Model Factories. Implement one-to-one, one-to-many, and many-to-many relationships between models. Perform database operations using Fluent Query Builder. Implement query scopes and model events in Laravel.	6
Module 3: Routing, Controllers and Blade Templates Create static and dynamic routes with route parameters. Develop route groups and named routes. Implement controllers and resource controllers for CRUD operations. Design responsive web pages using Blade template inheritance and layouts.	9

Develop reusable Blade components and partial views. Create HTML forms and process user input through controllers. Implement server-side validation using Form Request Validation. Display validation errors and success messages in Blade templates.	
Module 4: Authentication and Advanced Features Implement user registration, login, and logout functionality. Manage sessions, cookies, and flash messages. Develop middleware for route protection. Implement role-based authorization for different user categories. Implement pagination for database records. Upload, store, retrieve, and delete files using Laravel File Storage.	6
Module 5: REST APIs, Security and Deployment Develop RESTful APIs for CRUD operations. Exchange data in JSON format using API Resources. Implement token-based authentication using Laravel Sanctum. Demonstrate CSRF protection, SQL injection prevention, and XSS prevention. Implement exception handling and application logging. Optimize application performance using configuration, route, and view caching commands. Deploy a Laravel application on a local/production server following deployment best practices.	9

Text Books:

1. Matt Stauffer, Two Scoops of Django 3.x: Best Practices for the Django Web Framework, Two Scoops Press, 2021.
2. Luke Welling and Laura Thomson, PHP and MySQL Web Development, 5th Edition, Addison-Wesley Professional, 2016.
3. Jesse Griffin, Laravel: Build Modern PHP Applications, Packt Publishing, 2024.

Reference Books:

1. Matt Zandstra, PHP Objects, Patterns, and Practice, 5th Edition, Apress, 2016.
2. Mark Masse, REST API Design Rulebook, O'Reilly Media, 2011.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	**% Distribution**
Performance	30
Quiz	10
Total	40

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1	3	0	0	0	1
CO2	3	3	3	3	1	0	0	1
CO3	3	3	3	3	2	0	1	1
CO4	2	2	3	3	2	0	3	1
CO5	2	3	3	3	3	2	3	2

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD5, CD6, CD7
CO2	CD1, CD2, CD4, CD5
CO3	CD1, CD2, CD3, CD4, CD5
CO4	CD1, CD2, CD3, CD4, CD6
CO5	CD2, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26652

Course Title: Full Stack using React JS Laboratory

Pre-requisite(s): Web Application Development using HTML and CSS

Co-requisite(s): Frontend Engineering with React JS

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Lopamudra Hota, Dr. Subrajeet Mohapatra

COURSE OBJECTIVES

This course enables the students to:

1. Develop reusable frontend components using React and JSX.
2. Design interactive user interfaces using React Hooks and component-based architecture.
3. Build Single Page Applications (SPA) using React Router and RESTful APIs.
4. Apply testing, debugging, and performance optimization techniques for React applications.
5. Develop and deploy industry-standard frontend applications following modern software engineering practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Develop React applications using JSX, functional components, props, and state.
CO2	Build interactive web applications using Hooks, forms, event handling, and reusable components.
CO3	Design Single Page Applications with routing, Context API, and REST API integration.
CO4	Test, debug, optimize, and manage React applications using modern frontend development tools.
CO5	Develop, document, and deploy a complete frontend project using industry-standard practices.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: React Fundamentals and Environment Setup Installation and configuration of Node.js, npm, and Vite, Create the first React application, Design reusable React components using JSX, Develop a student profile application using props.	3
Module 2: React State and Event Handling Implement Counter Application using useState, Design a To-Do List Application, Develop a Registration Form using controlled components, Implement dynamic rendering using lists and conditional rendering, Develop custom Hooks for reusable logic.	3
Module 3: Building Data-Driven React Applications Design a multi-page SPA using React Router, Implement navigation with nested routes, Fetch data from REST APIs using Fetch API, Consume REST APIs using Axios, Develop a Weather Dashboard using public APIs, Implement Context API for application-wide state management.	3
Module 4: Advanced React Development and Performance Optimization Develop a Shopping Cart using useReducer, Optimize application performance using React.memo, useMemo, and useCallback, Implement Lazy Loading and Suspense, Create Error Boundaries for exception handling, Perform Unit Testing using Jest and React Testing Library.	3
Module 5: Deployment and Capstone Project Design responsive interfaces using Tailwind CSS, Implement Login Authentication interface, Optimize production build, Deploy React application on Vercel/Netlify. Mini Project: Develop a complete React-based frontend application (such as an E-commerce Portal, Student Management System, Library Management System, or Online Course Portal).	3

Text Books:

1. Alex Banks, Eve Porcello, Learning React, O'Reilly Media, 2024.

Reference Books:

1. Stoyan Stefanov, React Up & Running, O'Reilly Media, 2021.
2. Anthony Accomazzo, Full Stack React: The Complete Guide to ReactJS, Fullstack.IO, 2017.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	3	1	0	0	2
CO2	2	2	3	3	2	0	0	2
CO3	2	3	3	3	2	1	0	2
CO4	2	3	3	3	2	1	1	3
CO5	2	3	3	3	3	2	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD5, CD6, CD7
CO2	CD1, CD2, CD4, CD5, CD6
CO3	CD1, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD5, CD6, CD7
CO5	CD2, CD4, CD5, CD6

TRACK: Advanced Modern Computing

COURSE INFORMATION SHEET

Course Code: CA26653

Course Title: Agentic AI Systems

Pre-requisite(s): Machine Learning, Reinforcement Learning

Co-requisite(s): Agentic AI Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Debjani Mustafi, Dr. Sudip Kumar Sahana

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental concepts, architecture, and lifecycle of Agentic AI systems.
2. Explain the reasoning, planning, memory, and tool utilization mechanisms in autonomous AI agents.
3. Analyze Retrieval-Augmented Generation (RAG), agent orchestration, and multi-agent collaboration techniques.
4. Explore modern Agentic AI frameworks, including LangChain, LangGraph, and Model Context Protocol (MCP).
5. Evaluate the applications, ethical considerations, security, and governance of Agentic AI systems.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the architecture, components, and lifecycle of Agentic AI systems. (BL2)
CO2	Apply reasoning, planning, memory, and tool integration concepts to autonomous AI agents. (BL3)
CO3	Analyze Retrieval-Augmented Generation (RAG), vector databases, and multi-agent collaboration for intelligent task execution. (BL4)
CO4	Examine Agentic AI frameworks, including LangChain, LangGraph, and MCP, for designing AI-driven workflows. (BL4)
CO5	Evaluate the performance, security, ethical considerations and future directions of Agentic AI Systems. (BL5)

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Agentic AI Traditional AI Vs Generative AI Vs Agentic AI, Fundamentals of intelligent agents, Characteristics of autonomous AI agents, Agent architectures: Reactive, Deliberative, Hybrid, and BDI (Belief-Desire-Intention), Components of Agentic AI: Perception, Reasoning, Planning, Memory, Action, Feedback, Agent lifecycle, Applications of Agentic AI	8
Module 2: Reasoning, Planning and Memory in Agentic AI LLMs as reasoning engines, Prompt engineering, Prompting techniques: Zero-shot, Few-shot, Chain-of-Thought (CoT), ReAct, Task planning and decomposition, Tool usage and function calling, Short-term and long-term memory, Context management, Agent execution loop	8
Module 3: Knowledge-Augmented Agents and Multi-Agent Systems Introduction to Retrieval-Augmented Generation (RAG), Embeddings and vector databases, Document indexing and semantic retrieval, Agentic RAG architecture, Introduction to multi-agent systems, Agent communication and collaboration, Task delegation and coordination, Applications of collaborative AI agents	8
Module 4: Agentic AI Frameworks and Workflow Orchestration Overview of Agentic AI development frameworks, LangChain architecture and components, LangGraph for graph-based workflows, Model Context Protocol (MCP): Concepts and architecture, Tool integration using MCP, Workflow orchestration, AI agents interacting with APIs and external services.	8
Module 5: Responsible Agentic AI Evaluation metrics for AI agents, Performance optimization, AI safety and security, Ethical issues in autonomous AI, Explainability and transparency, Privacy, governance, and regulatory considerations, Human-in-the-loop systems.	8

Text Books:

1. Roberto Infante, 2026, AI Agents and Applications, O’Reilly, Manning Publications Co., ISBN: 978-633436541

Reference Books:

1. Jay Alammar & Maarten Grootendorst, Hands-On Large Language Models, O'Reilly, 2024

2. Ken Huang (Ed.), Agentic AI: Theories and Practices, Springer, 2025.

3. Kaushik Bar, The Agentic AI Hand Book: Concepts, Design Patterns and Future Directions, 2025, ISBN: 9798265909534.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lectures by use of boards/LCD projectors/OHP projectors
CD2	Concept based Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Interactive Lectures, Flipped Classroom Activities
CD6	Self- learning through NPTEL materials and internets
CD7	Case Studies via Research Papers

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	0	1
CO2	3	3	2	2	0	0	0	1
CO3	2	3	3	2	1	0	0	2
CO4	1	2	3	3	2	1	0	2
CO5	0	1	1	2	1	1	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6
CO2	CD1, CD2, CD3, CD5, CD6, CD7
CO3	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO4	CD1, CD2, CD3, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26655

Course Title: Prompt Engineering

Pre-requisite(s): Python Programming, Natural Language Processing

Co-requisite(s): Prompt Engineering Lab

Credits: 03 L: 03 T: 0 P: 0

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Subrajeet Mohapatra, Dr. S. P. Singh, Dr. Kanchan Jha

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals of Generative AI, Large Language Models (LLMs), and prompt engineering
2. Design and optimize prompts using various prompting techniques to effectively interact with LLMs
3. Apply prompt engineering techniques for content generation, code assistance, and problem-solving tasks
4. Develop AI-assisted solutions by leveraging LLM APIs, prompt templates, and workflow automation
5. Evaluate the quality, reliability, and ethical aspects of AI-generated responses while promoting responsible AI practices

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts of Generative AI, Large Language Models (LLMs), and prompt engineering, and create effective basic prompts for AI applications
CO2	Apply various prompting techniques to generate structured outputs and improve the quality of AI-generated responses for different tasks
CO3	Utilize Large Language Models for content generation, code development, debugging, documentation, and multi-step problem solving through prompt engineering
CO4	Develop AI-powered applications by integrating LLM APIs with Python using prompt templates and reusable workflows
CO5	Evaluate the performance, reliability, and ethical implications of AI-generated responses, and apply responsible AI practices to mitigate issues such as hallucinations and bias.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Generative AI and Prompt Engineering Familiarization with Generative AI tools (ChatGPT, Gemini, Claude, Microsoft Copilot), Understanding Large Language Models (LLMs), Tokens, Context Windows, and Prompt Components, Creating and evaluating basic prompts, Prompt refinement through iterative experimentation	8
Module 2: Prompting Techniques and Structured Outputs Zero-shot, One-shot, and Few-shot prompting, Role-based and instruction-based prompting, Chain-of-thought and step-by-step prompting, Generating structured outputs in Markdown, JSON, XML, and tabular formats, Comparative evaluation of different prompting strategies	8
Module 3: AI-Assisted Content and Code Generation Content creation: emails, reports, summaries, and technical documentation, Code generation using LLMs (Python programs), Code explanation, debugging, optimization, and documentation, Data analysis and visualization assistance using prompts, Prompt chaining for multi-step problem solving	8
Module 4: LLM APIs and Application Development Introduction to OpenAI and Google Gemini APIs, API authentication and Python integration, Prompt templates and reusable workflows, Developing simple AI-powered applications (chatbots, document summarizers, question-answering systems, Basic introduction to frameworks such as LangChain	8
Module 5: Responsible AI, and Evaluation Prompt evaluation and performance comparison, Hallucination detection and mitigation techniques, Ethical, legal, and responsible use of Generative AI	8

Text Books:

1. James Phoenix and Mike Taylor, Prompt Engineering for Generative AI, O'Reilly Media, 2024.
2. Jay Alammar and Maarten Grootendorst, Hands-On Large Language Models, O'Reilly Media, 2024.

Reference Books:

1. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, 2022.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	% Distribution
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	0	2	0	0	0	2

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO2	2	3	2	3	0	0	0	2
CO3	2	3	3	3	1	0	0	2
CO4	2	2	3	3	2	1	0	2
CO5	1	2	2	2	1	0	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD6, CD7
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD4, CD5, CD6
CO4	CD1, CD4, CD5, CD6
CO5	CD1, CD2, CD3, CD4, CD6, CD7

<u>**COURSE INFORMATION SHEET**</u>

Course Code: CA26657

Course Title: Quantum Computing

Pre-requisite(s): Mathematics for Computing

Co-requisite(s): Quantum Computing Lab

Credits: 3 L: 3 T:0 P: 0

Class Schedule per Week: 3

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Rathindranath Dutta, Dr. Kuntal Mukherjee

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamental principles of quantum mechanics and their role in quantum computing
2. Analyze quantum states, qubits, quantum gates and quantum circuits
3. Design quantum corcuits.
4. Design efficient quantum algorithms.
5. Apply quantum commutating in real-life applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain quantum mechanics and its relevance to quantum computation.
CO2	Analyze qubits, quantum states, measurements and quantum circuits.
CO3	Design and analyze quantum corcuits.
CO4	Design and analyze quantum algorithms.
CO5	Understand real-life applications of quantum commutating and realization of practical quantum machines.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Quantum Mechanical Foundations Overview of quantum computing and some notable applications; introduction to quantum physics and quantum mechanics, Wave-particle duality, uncertainty principle, postulates of Quantum Mechanics, Concepts of wave function: Schrödinger equation, Concept of superposition, Vector formulation of quantum mechanics: Dirac Formulation.	8
Module 2: Mathematical Foundations and Quantum Model of Computation Qubits and computational basis states; state vectors, superposition, probability amplitudes; Dirac (Bra–Ket) notation; inner, outer, and tensor products; normalization, projection operators, density matrices; Bloch sphere representation, quantum measurement, and bipartite and multipartite quantum systems.	8
Module 3: Quantum Circuits and Quantum Supremacy Quantum circuit model, quantum gates, single-qubit and multi-qubit quantum gates, matrix representation of quantum gates, quantum circuits, CNOT, SWAP, Toffoli and Fredkin gates, quantum logic design, quantum parallelism, quantum supremacy, no-cloning theorem, oracle-based computation, fundamentals of ZX-Calculus and graphical representation of quantum circuit	8
Module 4: Quantum Algorithms and Programming Quantum entanglement, Bell states, quantum teleportation, superdense coding, analysis of Deutsch and Deutsch–Jozsa algorithms, Overview of Quantum Fourier Transform, Shor's, Simon's, Quantum Phase Estimation, Grover's algorithms; Comparison between classical and quantum algorithms.	8
Module 5: Applications and Emerging Technologies Applications of quantum computing in cryptography, machine learning. Different realization techniques of quantum computing hardware. Notion of error correction in quantum computing and fault-tolerant quantum machines.	8

Text Books:

1. Quantum Computation and Quantum Information, Michael A. Nielsen and L. Chuang, 10th Anniversary Edition, 2010, Cambridge University Press, **ISBN-13: 978-1107002173**
2. Introduction to Quantum Computing:A Practical Approach Using Python, Nikhil Ranjan Roy and K.Mukherjee, 2nd Edition, 2026, S.Chand Publication, ISBN:9789373590745

Reference Books:

1. Quantum Computing: From Linear Algebra to Physical Realizations, Mikio Nakahara and Tetsuo Ohmi, CRC Press, 2008, ISBN-13: 978-0750309837
2. *Quantum Computing for Computer Scientists,* Yanofsky, N. S., and Mannucci, M. A.,2008, Cambridge University Press, Cambridge, UK. ISBN-13: 978-0521879965.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
End Semester Examination	50

Continuous Internal Assessment	**% Distribution**
3 Quizzes (Best two Quizzes)	30
Assignment(s)	10
Seminar before a committee	10
Total	50

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Classroom Lectures
CD2	PPT Presentations
CD3	Problem Solving Sessions
CD4	Qiskit Demonstrations, Cirq Demonstration
CD5	Seminars
CD6	Assignments

Code	Course Delivery Method
CD7	Mini Project

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	0	0	0	2
CO2	3	3	2	2	0	0	0	2
CO3	3	3	3	3	1	0	0	2
CO4	3	3	3	3	1	1	0	3
CO5	2	2	2	3	1	1	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD3, CD6
CO2	CD1, CD2, CD3, CD4, CD6
CO3	CD1, CD3, CD4, CD6, CD7
CO4	CD1, CD2, CD4, CD5, CD6, CD7
CO5	CD1, CD2, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26654

Course Title: Agentic AI Systems Lab

Pre-requisite(s): Python Programming, Basics of Machine Learning

Co-requisite(s): Agentic AI

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Debjani Mustafi, Dr. Poonam Moral

COURSE OBJECTIVES

This course enables the students to:

1. Develop basic AI agents using Large Language Models (LLMs) and prompt engineering techniques.
2. Build agentic workflows using LangChain and implement memory-enabled AI agents.
3. Design Retrieval-Augmented Generation (RAG) pipelines for knowledge-grounded AI applications.
4. Construct graph-based multi-agent workflows using LangGraph and integrate external tools through MCP.
5. Evaluate, debug, and deploy agentic AI applications using modern development frameworks and best practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Build Simple LLM-based AI agents using prompt engineering and agent frameworks. (BL3)
CO2	Implement memory-enabled AI agents using LangChain with short-term and long-term memory mechanisms. (BL3)
CO3	Develop Retrieval-Augmented Generation (RAG) applications using vector databases and document retrieval techniques. (BL6)
CO4	Create graph-based and multi-agent workflows using LangGraph and integrate external tools through the Model Context Protocol (MCP). (BL6)
CO5	Evaluate, optimize, and deploy agentic AI applications using standard testing and monitoring practices. (BL6)

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Agentic AI Development Setting up Python Environment for Agentic AI, OpenAI-compatible LLM APIs, Building a simple conversational AI agent, Function calling and structured outputs, Tool-enabled chatbot development, Testing and debugging AI agents. **Tools:** Python, Jupyter Notebook, OpenAI API/Ollama, VS Code.	4
Module 2: LangChain and AI Memory Building chains and prompts, Implementing Sequential Chains, Implementing Sequential Chains, Short-term memory using conversation buffers, Long-term memory using persistent storage, Managing conversational context. **Tools:** LangChain, FAISS/ChromaDB, SQLite.	9
Module 3: Retrieval-Augmented Generation (RAG) Vector embeddings, Document loading and preprocessing, Text chunking strategies, Vector databases, Similarity search, Building a basic RAG chatbot, Improving response quality using retrieved context, Evaluation of RAG responses. **Tools:** LangChain, ChromaDB, FAISS, Sentence Transformers.	9
Module 4: LangGraph, Multi-Agent Systems and MCP Creating graph-based workflows, State management in LangGraph, Multi-agent collaboration, Task routing between agents, Working with Model Context Protocol (MCP), Connecting AI agents with external tools and APIs through MCP, Building a workflow involving multiple specialized agents. **Tools:** LangGraph, MCP SDK, LangChain.	9
Module 5: Agent Deployment, Evaluation and Mini Project Testing AI agents, Monitoring agent execution, Prompt and Memory optimization, Safety guardrails, Basic deployment using Streamlit/Gradio, Performance evaluation, Mini projects such as AI Research Assistant, Customer Support Agent, Multi-agent Task Planner etc.	9

Text Books:

1. V. Alto, Building LLM Powered Applications: Create intelligent apps and agents with large language models. Packt Publishing Ltd., 2024

2. B. Auffarth, L. Kuligin, Generative AI with LangChain- Second Edition. Packt Publishing, 2025

Reference Books:

1. Biswas and W. Talukdar, Building Agentic AI Systems: Create intelligent, autonomous AI agents that can reason, plan, and adapt. Packt Publishing Ltd., 2025

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	**% Contribution during CO Assessment**
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	**% Distribution**
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	**% Distribution**
Performance	30
Quiz	10
Total	40

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	**Course Delivery Method**
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)

Code	Course Delivery Method
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	3	1	1	1	2
CO2	2	2	3	3	2	1	1	2
CO3	2	3	3	3	1	1	1	2
CO4	2	3	3	3	2	1	2	3
CO5	2	3	3	3	2	1	2	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6, CD7
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD4, CD5, CD6
CO4	CD1, CD2, CD4, CD5, CD6, CD7
CO5	CD2, CD3, CD4, CD5, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26656

Course Title: Prompt Engineering Lab

Pre-requisite(s): Programming in Python

Co-requisite(s): Prompt Engineering

Credits: 1.5 L: 0 T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Kanchan Jha, Dr. Subrajeet Mohapatra

COURSE OBJECTIVES

This course enables the students to:

1. Understand the fundamentals of Generative AI, Large Language Models (LLMs), and prompt engineering.
2. Design and optimize prompts using various prompting techniques to effectively interact with LLMs.
3. Apply prompt engineering techniques for content generation, code assistance, and problem-solving tasks.
4. Develop AI-assisted solutions by leveraging LLM APIs, prompt templates, and workflow automation.
5. Evaluate the quality, reliability, and ethical aspects of AI-generated responses while promoting responsible AI practices.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Explain the fundamental concepts of Generative AI, Large Language Models (LLMs), and prompt engineering, and create effective basic prompts for AI applications.
CO2	Apply various prompting techniques to generate structured outputs and improve the quality of AI-generated responses for different tasks.
CO3	Utilize Large Language Models for content generation, code development, debugging, documentation, and multi-step problem solving through prompt engineering.
CO4	Develop AI-powered applications by integrating LLM APIs with Python using prompt templates and reusable workflows.
CO5	Evaluate the performance, reliability, and ethical implications of AI-generated responses, and apply responsible AI practices to mitigate issues such as hallucinations and bias.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction to Generative AI and Prompt Engineering Familiarization with Generative AI tools (ChatGPT, Gemini, Claude, Microsoft Copilot), Understanding Large Language Models (LLMs), Tokens, Context Windows, and Prompt Components, Creating and evaluating basic prompts, Prompt refinement through iterative experimentation	9
Module 2: Prompting Techniques and Structured Outputs Zero-shot, One-shot, and Few-shot prompting, Role-based and instruction-based prompting, Chain-of-thought and step-by-step prompting, Generating structured outputs in Markdown, JSON, XML, and tabular formats, Comparative evaluation of different prompting strategies	9
Module 3: AI-Assisted Content and Code Generation Content creation: emails, reports, summaries, and technical documentation, Code generation using LLMs (Python programs), Code explanation, debugging, optimization, and documentation, Data analysis and visualization assistance using prompts, Prompt chaining for multi-step problem solving	9
Module 3: LLM APIs and Application Development Introduction to OpenAI and Google Gemini APIs, API authentication and Python integration, Prompt templates and reusable workflows, Developing simple AI-powered applications (chatbots, document summarizers, question-answering systems, Basic introduction to frameworks such as LangChain	6
Module 5: Responsible AI, and Evaluation Prompt evaluation and performance comparison, Hallucination detection and mitigation techniques, Ethical, legal, and responsible use of Generative AI	6

Text Books:

1. James Phoenix and Mike Taylor, Prompt Engineering for Generative AI, O'Reilly Media, 2024.

2. Jay Alammar and Maarten Grootendorst, Hands-On Large Language Models, O'Reilly Media, 2024.

Reference Books:

1. Lewis Tunstall, Leandro von Werra, and Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, 2022.

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	0	2	0	0	0	2
CO2	2	3	2	3	0	0	0	2
CO3	2	3	3	3	1	0	0	2
CO4	2	2	3	3	2	1	0	2
CO5	1	2	2	2	1	0	3	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	**Course Delivery Method**
CO1	CD1, CD2, CD6, CD7
CO2	CD1, CD2, CD3, CD6
CO3	CD1, CD2, CD4, CD6, CD7
CO4	CD1, CD2, CD4, CD6
CO5	CD1, CD2, CD3, CD4, CD6, CD7

COURSE INFORMATION SHEET

Course Code: CA26658

Course Title: Quantum Computing Lab

Pre-requisite(s): Mathematics for Computing, Python Programming

Co-requisite(s): Quantum Computing

Credits: 1.5 L:0 0T: 0 P: 03

Class Schedule per Week: 03

Class: MCA

Semester / Level: III / VI

Branch: MCA

Name of the Teachers: Dr. Kuntal Mukherjee, Dr. Akriti Nigam

COURSE OBJECTIVES

This course enables the students to:

1. Understand quantum software frameworks for simulation.
2. Simulate quantum gates and quantum states.
3. Implement quantum circuits and analyze their behavior.
4. Implement quantum algorithms and analyze their behavior.
5. Perform experiments on with real applications.

COURSE OUTCOMES (COs)

After the completion of this course, students will be able to:

CO1	Configure quantum computing software environments.
CO2	Design and simulate quantum gates and states.
CO3	Design and simulate quantum circuits.
CO4	Implement and analyze fundamental quantum algorithms.
CO5	Design and implement real quantum computing algorithms.

SYLLABUS

MODULE	NO. OF LECTURE HOURS
Module 1: Introduction and Environment Setup Installation of Qiskit / Cirq / PyQuil with Jupyter Notebook.	3
Module 2: Representation of Single-Qubit Quantum States Represent Computational Basis States, Construct and Normalize Single-Qubit Quantum States. Generate Computational Basis and Superposition States, Generate Computational Basis and Superposition States, Visualization of Single-Qubit Quantum States on the Bloch Sphere.	12
Module 3: Quantum Gates and Circuit Operations Implementation of Single-Qubit Quantum Gates (X, Y, Z, H, S, T), Bloch Sphere Representation and State Visualization, Implementation of Multi-Qubit Gates (CNOT, SWAP, Toffoli, Fredkin).	9
Module 4: Implementation of Quantum Algorithms Implementation of Deutsch Algorithm, Implementation of Deutsch–Jozsa Algorithm, Implementation of Grover's Search Algorithm, Implementation of Quantum Fourier Transform.	9
Module 5: Applications of Quantum Computing Implementation of real quantum algorithms and simulation of quantum error correcting codes.	6

Text Books:

1. Mastering Quantum Computing with IBM QX: Explore the World of Quantum Computing Using the Quantum Composer and Qiskit, Christine Corbett Moran,1st Edition, Packt Publishing, 2019, ISBN: 978-1789136432.
2. Introduction to Quantum Computing: A Practical Approach Using Python, Nikhil Ranjan Roy and K. Mukherjee, 2nd Edition, 2026, S. Chand Publication, ISBN:9789373590745.

Reference Books:

1. Quantum Computation and Quantum Information, Michael A. Nielsen and L. Chuang, 10 th Anniversary Edition, 2010, Cambridge University Press, ISBN-13: 978-1107002173

2. Quantum Computing: From Linear Algebra to Physical Realizations, Mikio Nakahara and Tetsuo Ohmi, CRC Press, 2008, ISBN-13: 978-0750309837

COURSE OUTCOME (CO) ATTAINMENT ASSESSMENT TOOLS & EVALUATION PROCEDURE

Direct Assessment

Assessment Tool	% Contribution during CO Assessment
Continuous Internal Assessment	60
End Semester Examination	40

Continuous Internal Assessment	% Distribution
Day-to-day performance and Lab file	30
Quiz(es)	10
Viva	20
Total	60

End Semester Exam	% Distribution
Performance	30
Quiz	10
Total	40

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	✓	✓	✓	✓	✓
Semester End Examination	✓	✓	✓	✓	✓

Indirect Assessment

- Student Feedback on Course Outcome

Course Delivery Methods

Code	Course Delivery Method
CD1	Lecture by use of boards/LCD projectors
CD2	Tutorials/Assignments
CD3	Seminars/ Quiz (s)
CD4	Mini projects/Projects
CD5	Laboratory experiments/teaching aids
CD6	Online Materials
CD7	Video Lectures for Self-Studies

Mapping of Course Outcomes onto Program Outcomes and Program Specific Outcomes

Grading: 0**-** No Correlation **(0)**, 0-44.99- Low Correlation **(1)**, 45-64.99 Medium Correlation **(2)**, 65-100 High Correlation **(3)**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	3	1	0	0	2
CO2	3	3	2	3	1	0	0	2
CO3	3	3	3	3	2	1	0	2
CO4	3	3	3	3	2	1	1	3
CO5	3	3	3	3	2	2	1	3

Mapping Between Course Outcomes and Course Delivery Method

Course Outcomes	Course Delivery Method
CO1	CD1, CD2, CD5, CD6, CD7
CO2	CD1, CD2, CD5, CD6
CO3	CD1, CD2, CD4, CD5, CD6
CO4	CD1, CD2, CD4, CD5, CD6
CO5	CD2, CD4, CD5, CD6, CD7

*****