Curriculum Framework

Master of Computer Application

As per NEP 2020 and Learning Outcomes-based National Curriculum Framework

(Aligned with NCrF and NHEQF)

Effective From Academic Year 2026-2027

GUJARAT VIDYAPITH: AHMEDABAD

Curriculum Framework of Master of Computer Application

Published by:

Dr. Himanshu Patel

Registrar

Gujarat Vidyapith

Near Income Tax Office, Ashram Road, Ahmedabad - 380009.

From the Desk of Vice Chancellor…

Dear All,

Any curriculum, at any level, must be firmly grounded in the objectives and goals that an educator or an educational institution aspires to achieve for its students. A course that trains students to solve mathematical equations must be very different from one that teaches them how to play a musical instrument, practice yoga, or conduct social research. Each subject requires its own methods, activities, and learning goals, which is why curriculum design is so important.

Therefore, curriculum is of utmost importance, as it determines how teachers and students will spend their time—in laboratories, in clinical practice, in creative endeavors, or in interactive lectures. It also reflects what a class, a department, a school, or an institution values; what it defines as its mission; and what it expects its graduates to accomplish. In this sense, the curriculum is the map that guides the essentials of learning from the classroom level to the institutional level.

The true success of any curriculum must be judged by its ability to achieve its intended objectives. It is a test of how well an institution—or an individual teacher—understands and articulates those objectives, and how effectively a pathway is created for students to attain success as defined by them. Curriculum is, in fact, a continuous chain of activities designed to translate broad educational goals into concrete practices, learning materials, and observable changes in behavior. A lesson plan, for instance, is curriculum at the classroom level, answering the critical questions: *What do I want my students to know? How can I engage them meaningfully? How will I measure what they have learned?*

For a society to achieve its educational aspirations, the curriculum must be both functional and relevant to its needs. Through careful management of curriculum, effective use of resources, and policies that bring systemic improvements, education can move society toward a more promising future. Indeed, curriculum is the very foundation of any academic institution—without it, the institution would lose its purpose and direction.

At Gujarat Vidyapith, established by Mahatma Gandhi in 1920 with its rich cultural and educational heritage, we remain committed to these ideals and we work with well-defined objectives to prepare our students for a brighter academic and social future.

It gives me immense pride and joy to announce the unveiling of the latest curriculum of Gujarat Vidyapith. This curriculum has been carefully designed in alignment with the objectives and guiding principles of the National Education Policy (NEP) 2020. I take this opportunity to place on record my deep appreciation for the efforts of the teaching faculty of Gujarat Vidyapith, the Members of the Board of Studies, and the Members of the Academic Council. Their dedication and vision have given shape to this comprehensive neo-curriculum, which will guide our students and our institution toward continued excellence.

With best wishes,

Dr. Harshad Patel
Vice Chancellor
Gujarat Vidyapith

Curriculum Framework

Master of Computer Application

Effective From Academic Year 2026-2027

Department of Computer Science

Faculty of Information, Communication and Technology

Gujarat Vidyapith

Board of Studies

Chairperson:

Prof. Ajay Parikh
Professor & Dean, Faculty of Information, Communication and Technology, Gujarat Vidyapith, Ahmedabad.

External Experts:

1) **Prof. Rajeev Wankar**
   Professor, University of Hyderabad

2) **Prof. Nilesh Modi**
   Professor, Babasaheb Ambedkar Open University

Members from the Department:

1) **Prof. Dhiren Patel**
   Professor, Faculty of Information, Communication and Technology, Gujarat Vidyapith, Ahmedabad.

2) **Dr. Neepa Shah**
   Associate Professor, Faculty of Information, Communication and Technology, Gujarat Vidyapith, Ahmedabad.

3) **Mr. Kamaleshkumar Salunke**
   Assistant Professor, Faculty of Information, Communication and Technology, Gujarat Vidyapith, Ahmedabad.

4) **Dr. Ruchita Shah**
   Assistant Professor, Faculty of Information, Communication and Technology, Gujarat Vidyapith, Ahmedabad.

PROGRAMME STRUCTURE							
Course Code	Course Name	Hours			Credit	Evaluations	
		Theory	Practical	Total		CCE	TEE
SEMESTER-1							
266010345032	Object Oriented Programming using Java	45	0	45	3	40	60
266010445033	Data Structures	60	0	60	4	40	60
266010245034	Full Stack Web Development	30	0	30	2	40	60
266010345035	Advance Database Management System	45	0	45	3	40	60
266010245036	Lab based on Object Oriented Programming using Java	0	60	60	2	40	60
266010245037	Lab based on Data Structures	0	60	60	2	40	60
266010245038	Lab Based on Full Stack Web Development	0	60	60	2	40	60
266010245039	Lab based on Advanced Database Management System	0	60	60	2	40	60
-	Community Living	-	60	60	2*	Grade	Grade
Total		**180**	**300**	**480**	**20+2***	**320**	**480**
SEMESTER-2							
256010345009	Operating System- Elective-I	45	0	45	3	40	60
256010345010	Mobile Application Development-Elective-II						
256010345011	Web Technology (Lab based)	0	90	90	3	40	60
256010445012	Computer Network	60	0	60	4	40	60
256010345013	Software Design Pattern (GOF)	45	0	45	3	40	60
256010345014	Software Engineering	45	0	45	3	40	60
256010245015	Lab based on Operating System	0	60	60	2	40	60
256010245016	Lab based on Mobile Application Development						
256010245017	Lab based on Software Design Pattern (GOF)	0	60	60	2	40	60
-	Community Living	-	60	60	2*	Grade	Grade
Total		**195**	**270**	**465**	**20+2***	**280**	**420**
SEMESTER-3							
266510245040	Microservices using Spring Boot	30	0	30	2	40	60
266510245041	Cloud Computing	30	0	30	2	40	60
266510345042	Blockchain Technology-E1	45	0	45	3	40	60
266510345043	Agentic AI & Autonomous Intelligent Systems-E2						
266520345053	MOOC-Programming with Generative AI-E3-Appendix-A						
266510345044	Advanced Database Management System-E1	45	0	45	3	40	60
266520345045	MOOC-Big Data Computing-E2-Appendix-B						
266510245046	Lab Based on Microservices using Spring Boot	0	60	60	2	40	60
266510145047	Lab Based on Cloud Computing	0	30	30	1	40	60
266510245048	Lab Based on Blockchain Technology-E1	0	60	60	2	40	60
266510245049	Lab Based on Agentic AI & Autonomous Intelligent Systems -E2						
266510245054	MOOC-Lab Based on Programming with Generative AI-E3-Appendix-C						
266510145050	Lab Based on Advance Database Management System-E1	0	30	30	1	40	60
266510145051	Case Study Based on MOOC-E2-Appendix-D						
266510445052	Internship/Project	0	120	120	4	40	60
-	Community Living	-	60	60	2*	Grade	Grade
Total		**150**	**360**	**510**	**20+2***	**360**	**540**
SEMESTER-4							
266512045032	Internship	0	600	600	20	40	60
Total		**0**	**600**	**600**	**20**	**120**	**180**
GRAND TOTAL		**525**	**1530**	**2055**	**40**	**1080**	**1620**

***CCE- Continuous Comprehensive Evaluation; **TEE- Term End Evaluation**

Programme Contents	Credits
Core Courses	**56**
Internship	**24**
Total Credits of Programme	**80**

Collaborating Organisations

- **Tiny ERP (Odoo)**
- **Tecblic India**
- **Complitech Solution Pvt. Ltd.**
- **CMExpertise Infotech Pvt.Ltd**
- **Yudiz Solutions Pvt Ltd**
- **Qurious Tech**
- **PLUSINFOSYS**
- **Addon Solutions**
- **PerceptionCare**
- **Raimptech Solutions Pvt. Ltd**
- **Xomic infotech**
- **Confidosoft**
- **Serpent Consulting Pvt. Ltd.**
- **Maxgen Technologies Ltd.**
- **Vowels digitech software Pvt. Ltd**
- **IncDesign Agency LLP**
- **Esfina Technology & Solution Pvt. Ltd.**
- **Infoware**
- **Mday Consultancy**
- **MV Clouds Private Limited**
- **NUWAVE COMMUNICATION Inc.**
- **Emerging Five**
- **Neela Spaces Limited**
- **TRENTIUM SOLUTION PRIVATE LIMITED**
- **Echobitz IT Solutions Pvt Ltd**
- **Macdroid Solutions PVT. LTD.**
- **Restrosoft Solutions Pvt. Ltd.**
- **Cilans System**
- **Wibit Solutions LLP**
- **Whitestork software solutions**

Programme Outcomes (POs)

After successful completion of the MCA program, students will be able to:

PO-1	**Advanced Knowledge of Computer Application**	Equips students with in-depth knowledge of computer applications, including software development, database management, operating systems, computer network and cyber security. It emphasizes advanced concepts in computing to build efficient, scalable, and secure solutions. Students are prepared to tackle complex IT challenges across various industry domains.
PO-2	**Research & Problem-Solving Skills**	To inculcate the ability to undertake independent and collaborative research that leads to new knowledge, innovative IT solutions, and practical applications; and to nurture an entrepreneurial mindset for creating socially responsible start-ups that address local, rural, and national challenges in alignment with Gandhian principles of self-reliance, sustainability, and service to humanity.
PO-3	**Experimental & Analytical Proficiency**	Develop strong practical skills in programming, software development, and system design through hands-on experience with various languages, tools, and platforms. At the same time, enhance analytical and problem-solving abilities to effectively analyze complex challenges and design efficient, technology-driven solutions. This integrated approach prepares students to tackle real-world computing problems with confidence and competence.
PO-4	**Interdisciplinary Approach**	Integrate knowledge from various domains, disciplines, and societal contexts to develop inclusive, people-centric IT solutions that promote rural development and digital empowerment.
PO-5	**Environmental Consciousness & Sustainability**	Understand the environmental impact of IT systems, including energy use and e-waste. Apply green computing practices in software development, cloud services, and data centers. Design ICT-based solutions to address environmental and sustainability challenges. Promote ethical responsibility and compliance with environmental regulations in IT. Contribute to sustainable development goals through eco-conscious IT innovation.
PO-6	**Ethics & Professional Values**	Instil a sense of ethical and social responsibility in students, emphasizing the importance of ethical practices, and data privacy. Apply appropriate technology in the context of society, sustainability, environment and ethical practices.
PO-7	**Effective Technical Communication**	Communicate technical information effectively in Gujarati and English, enabling outreach to local communities as well as participation in national/global digital initiatives.
PO-8	**Modern Technological Applications**	Keep students abreast of the latest developments and trends in the IT industry, including emerging technologies.
PO-9	**Teamwork & Leadership in Research**	Enhances leadership and teamwork skills in the context of software and technology projects. Prepares students to collaborate effectively in diverse development teams and manage real-world IT tasks. Encourages collective problem-solving to address complex technological challenges.
PO-10	**Lifelong Learning & Adaptability**	Recognize the need for lifelong learning and actively engage in continuous upskilling through advanced courses, certifications, and research activities, in alignment with NEP's flexible learning approach.
PO-11	**Project Management & Entrepreneurial Thinking**	Develop entrepreneurial abilities to initiate and manage start-ups or social enterprises that offer innovative applications for local, rural, and national development.
PO-12	**Social & Community Engagement**	Use computing knowledge and skills to contribute positively to society, environment, and governance, focusing on eco-friendly solutions, digital inclusion, and nation-building aligned with the vision of Gram Swaraj.

Programme Specific Outcomes (PSOs)

After successful completion of the MCA program with a specialization in Computer Science & Application, students will be able to:

PSO-1	Apply advanced computing principles, algorithms, programming languages (such as Java, Python, C++) and Web Technology to design, develop, and analyse complex software systems and data-driven solutions for industrial and societal challenges.	Aligns with **PO1 (Advanced Knowledge of Computer Application), PO2 (Research & Problem-Solving Skills),** supporting rigorous academic and professional preparation as envisioned by Gujarat Vidyapith.
PSO-2	Employ modern tools, frameworks, and technologies—including machine learning, cloud computing, IoT, Cyber Security and data analytics—to solve real-world problems, foster digital innovation, provide security and promote rural-urban technology integration.	Supports **PO8 (Modern Technological Applications),** digital literacy, and reflects **NEP 2020's vision** for technology-driven education and development.
PSO-3	Demonstrate independent and collaborative work through research projects, industrial training, and interdisciplinary approaches that advance sustainability, simplicity, and social well-being in line with Gandhian philosophy.	Support **PO5 (Environmental Consciousness & Sustainability)** and aligns with Gujarat Vidyapith's mission of rural development, service, and value-based inquiry.
PSO-4	Communicate technical concepts effectively through presentations, reports, and professional interactions, upholding ethics, cultural sensitivity, and community engagement, with emphasis on Indian knowledge systems and local languages.	Connects with **PO6 (Ethics & Professional Values), PO7 (Effective Technical Communication), PO12 (Social & Community Engagement),** and promotes Gandhian values and regional language proficiency.

CO Attainment Levels (OBE & NEP 2020 Aligned)

COs Attainment Levels	**Level**	**Description**	**Attainment Criteria**			
	Level 3	High	≥ 80% of students score above a set performance benchmark			
	Level 2	Moderate	60%–79% of students meet the performance benchmark			
	Level 1	Low	40%–59% of students meet the performance benchmark			
	Level 0	Not Attained	< 40% of students achieve the desired learning outcome			
Target Attainment (Benchmark)	**Cos**		**CO-1**	**CO-2**	**CO-3**	**CO-4**
	Target Level (%)		60	60	60	60

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010345032	Name of Course Object Oriented Programming using Java	Compulsory
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Demonstrate proficiency in Java programming fundamentals and object-oriented programming concepts by designing and implementing programs using classes, objects, constructors, inheritance, polymorphism, abstraction, interfaces, and access control mechanisms.

CO-2: Apply Java packages and exception handling techniques to develop modular, secure, and robust applications using built-in and user-defined packages, access protection, and effective error and exception management strategies.

CO-3: Develop Java applications involving file handling, stream-based input/output operations, serialization, and multithreading concepts, including thread creation, synchronization, and inter-thread communication for concurrent programming.

CO-4: Utilize Java Collection Framework and Generics to design efficient, type-safe, and reusable applications by employing collection classes, iteration techniques, generic classes, methods, and wildcard mechanisms.

Detailed Syllabus

Unit-1. Introduction to Java

1.1. **Java History, Features, Environment, Program Structure and Programming Building Block:**
History, Features and Environment, Program Structure, Java Environment, Structure of Java Program- Class, main() method. Overview of Java programming constructs- Data Types, Variables & Constants, Operators, Precedence Rules and Associativity, Object Reference, Wrapper classes, Arrays and Strings, Type Conversion & Type Casting, Autoboxing, Unboxing, Command Line Argument, Control Statements.

1.2. **Object-Oriented Programming (OOP)**

1.2.1 Object-Oriented Programming Concepts: Need of object-oriented programming, Object Oriented Programming vs other Paradigms, Class and Objects, four pillars of OOPs (Encapsulation, Inheritance, Abstraction and Polymorphism)

1.2.2 Class: Definition, Creation.

1.2.3 Object: Definition, Creation, Instantiating and Initializing an Object.

1.2.4 Constructors: Declaration, Default & Parameterized constructor.

1.2.5 Garbage Collection and Finalization.

1.2.6 Methods: Declaration, Invocation & Overloading.

1.2.7 Encapsulation: Data hiding using private variables, Abstraction: Abstract class and methods

1.2.8 Access control/protection: Types of Access Specifiers- Public, Private, Protected, Default, Modifiers- static, final, abstract, synchronized, volatile

1.2.9 Inheritance: Types: Single, Multilevel, Hierarchical, super and final

1.2.10 Polymorphism: Types of polymorphism, Method Overloading (Compile-time), Method Overriding (Runtime)

1.2.11 Interface: Declaration, Need, implementation, Multiple inheritance using interface

Unit-2. Java Package, Exception Handling

1.1. **Java Package:**
Introduction, Advantages of package, Types of Packages- Built-in, User-defined, creating package (package keyword) and Using package (import keyword), Access Protection, Importing Package, Important Built-in Packages: java.lang, jav.util, jav.io, etc.

1.2. **Exception and Error in Java:**

1.2.1 Types of Errors: Compile-time, Runtime, Logical errors

1.2.2 Exception and Error

1.2.3 Types of Exceptions: Checked and Unchecked
1.2.4 Exception Handling Mechanism: try, catch, finally blocks Keywords- throw and throws.
1.2.5 User-defined Exception: Creating custom exception class, Exception Encapsulation and Enrichment

Unit-3 Java Stream, File Handling and Multithreading

1.1. **Java Streams, Input / Output and File Handling**
1.1.1 Streams: Definition, concepts, advantages
1.1.2 Types of streams: Byte Streams, Character Streams
1.1.3 Byte and Character stream class hierarchy
1.1.4 File class
1.1.5 File Operations: Read, write, copy files (both stream)
1.1.6 Serialization: definition, need and serialize class

1.2. **Multithreading and Concurrency**
1.2.1 Thread Concept: Introduction, Definition and advantages, Process Vs. Thread
1.2.2 Java Thread Model: Main Thread, Thread class, and Runnable interface, Thread priority
1.2.3 Thread Life Cycle Stages: New, Runnable, Running, Waiting, Dead
1.2.4 Thread Creation: By extending Thread class, by implementing Runnable interface
1.2.5 Thread Methods: sleep(), join(), start(), run() etc.
1.2.6 Synchronization: Synchronization Methods, Synchronization Blocks, Deadlock Race Condition
1.2.7 Inter-thread Communication**:** wait() , notify(), notifyAll()

Unit-4 Java Collection Framework and Generic

1.1. **Collection Framework**
1.1.1 Introduction to Introduction to Collection Framework and Java.util package Goals and Advantages of Collection Framework,
1.1.2 List Interface: ArrayList, LinkedList
1.1.3 Set Interface: HashSet, LinkedHashSet, TreeSet
1.1.4 Queue Interface: PriorityQueue, Deque
1.1.5 Map Interface: HashMap, TreeMap, Hashtable
1.1.6 Iteration Techniques: Iterator, ListIterator, Enhanced for-loop
1.1.7 Basic types of collections classes – ArrayList, LinkedList, Iterator, Enumeration, Vector, Stack

1.2. **Generic**
1.2.1 Introduction to Generics
1.2.2 Type Safety
1.2.3 Reusability
1.2.4 Generic Class and Generic Method
1.2.5 Wildcards Upper Bound (? extends), Lower Bound (? super), Unbounded (?)

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	1	1	2	1	2	1	–	1.42	3	2	1	1	1.75
CO-2	3	3	3	1	–	2	1	2	1	2	1	–	1.58	3	2	1	1	1.75
CO-3	3	3	3	1	1	2	1	3	2	2	2	1	2	3	3	2	1	2.25
CO-4	3	2	3	1	–	1	1	3	2	2	2	–	1.67	3	3	1	1	2
Avg	3	2.5	3	1	0.25	1.5	1	2.5	1.5	2	1.5	0.25		3	2.5	1.25	1	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Seminar, Problem Solving, Examples, Questioning, Experiment, Demonstration
CO-2 (Unit: 2)	
CO-3 (Unit: 3)	
CO-4 (Unit: 4)	

Assessment Method						
Continuous Comprehensive Evaluation 40 Marks	**COs**	**Marks**	**Exam Component**			
			Class Test	**Assignment**	**Presentation/ Seminar**	**Involvement**
	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Herbert Schildt - Java the Complete Reference, McGraw Hill Education
- E. Balaguruswamy, Programming with Java A Primer, Mc Graw Hill
- Herbert Schildt, Dale Skrien, Java Fundamentals A comprehensive introduction McGraw Hill Education.
- James Rumbaugh, Michael Blaha, Object-Oriented Modelling and Design, Publication - Phi.

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central)
- e-PGPathsala(e-PGPathshala (inflibnet.ac.in))

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010445033	Name of Course Data Structures	Compulsory
Credit: 04	Teaching Scheme: Theory (60) - Practical (0)	Teaching Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Analyze and implement fundamental linear data structures such as arrays, stacks, queues, and linked lists, including their operations and applications.

CO-2: Understand and apply binary search tree concepts, including tree traversals, insertion, deletion, and basic balanced tree mechanisms.

CO-3: Represent graphs using matrices and lists, perform graph traversals using BFS and DFS, and manage sparse matrices along with dynamic storage allocation techniques.

CO-4: Implement and compare various sorting and searching algorithms, and apply hashing techniques including collision resolution for efficient data retrieval.

Detailed Syllabus

Unit-1

1.1 **Introduction to Data Structure and Linear Data Structure:**Data Management concepts, Data types – primitive and non-primitive, Performance Analysis and Measurement (Time and space analysis of algorithms-Average, best and worst case analysis), Types of Data Structures- Linear & Non Linear Data Structures.

1.2 **Linear Data Structure**

Stack - Definition & Concept, Operations on stack, Applications of Stacks, Conversion from infix to postfix Queue - Definition & Concepts, Operations on queue, Types of queue, Circular queue, Applications of Stacks priority queues, Priority queue, Process queue

1.3 **Linked Linear List -** Sequential & linked allocation, their advantages and disadvantages, Singly linked list and perations on it, Double linked list and operations on it, Circular linked list and operations on it, Applications of linked list.

Unit-2

2.1 **Non-Linear Data Structure:** Binary Search tree - Definition and Concepts, Representation, Operation Like

2.2 **Traversals**: Inorder, Preorder, Postorder, Insertion and Deletion in BST , Searching.in BST

2.3 **Introduce** Abstract Data Types (ADT), generic data structures, amortized analysis, recursion and recurrence relations, C++ STL, and Java Collections Framework.

2.4 **Some balanced tree mechanism:** AVL tree, B tree, B+ Tree, Segment Tree, Red-Black Tree

Unit-3.

3.1. **Graph -** Matrix representation of graph, Adjacency matrix, Path matrix, Dijkstra's algorithm, MINIMAL algorithm, Adjacency list representation of graph, Breadth First Search, Depth First Search, Floyd-Warshall algorithm, Kruskal’s algorithm

3.2. **Multilinked structure -** Sparse matrix, Sequential & linked allocation of sparse matrix, Matrix addition using sparse matrix

Unit-4.

4.1 **Hashing techniques -** Hashing Functions, hashing techniques – Bloom filters & Consistent hashing

4.2 **Sorting methods -** Insertion Sort, Quick sort (Partition Exchange sort), Radix sort, Heap sort, Merge sort and Performance comparison of sorting method.

4.3 **Searching -** Linear (sequential Search), Binary Searching

Mapping Matrix of POs, PSOs, and COs

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	–	1	2	1	1	–	–	1.67	3	2	2	1	2
CO-2	3	2	3	–	–	–	1	2	1	1	–	–	1.58	3	2	2	1	2
CO-3	3	3	3	1	–	–	–	2	1	1	–	–	1.67	3	2	2	1	2
CO-4	3	3	3	–	–	–	–	3	1	1	1	–	1.83	3	2	2	1	2
Avg	3	2.5	3.0	0.5	–	–	0.5	2.25	1.0	1.0	0.25	–		3	2	2	1	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Classroom Teaching, Group Discussion, Assignment, Examples, Questioning

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Tremblay & Sorenson, An introduction to Data Structures with applications,Tata McGrawHill
- Aaron M. Tenenbaum, Data Structures using C, PHI
- Reema Thareja, Programming in C, BPB publication, Oxford Higher Education, 2015
- Rajesh Shukla, Data Structures using C & C++ ,Wiley 2015

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central), e-PGPathsala(e-PGPathshala (inflibnet.ac.in))

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010245034	Name of Course Full Stack Web Development	Compulsory
Credit: 02	Teaching Scheme: Theory (30) - Practical (0)	Teaching Hours: 30

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand the architecture and working principles of modern web applications and explain frontend and backend technologies used in full-stack development.

CO-2: Design responsive and interactive user interfaces using HTML5, CSS3, JavaScript, and React.js.

CO-3: Explain and develop basic server-side applications and REST APIs using Node.js, Express.js, and MongoDB.

CO-4: Analyze the workflow of complete full-stack applications and apply modern development tools for application integration and deployment

Detailed Syllabus

Unit-1. Web Fundamentals & Frontend Basics

1.1 Evolution of Web Applications
1.2 Internet & WWW
1.3 Client-Server Architecture
1.4 Three-Tier Architecture
1.5 Static vs Dynamic Websites
1.6 HTTP & HTTPS
1.7 Frontend, Backend & Database
1.8 VS Code IDE
1.9 Browser Developer Tools
1.10 Introduction to Git & GitHub
1.11 HTML5
1.12 CSS3
1.13 Responsive Web Design

Unit-2. Frontend Development using React.js

2.1 Modern JavaScript (ES6)
2.2 DOM Manipulation
2.3 Modules
2.4 React Architecture
2.5 React Project Structure
2.6 JSX
2.7 Components
2.8 Props
2.9 State
2.10 Event Handling
2.11 Forms
2.12 React Hooks
2.13 Routing
2.14 API Calls using Fetch
2.15 Frontend Build Process

Unit-3. Backend Development using Node.js

3.1 Introduction to Node.js

3.2 npm
3.3 Express.js
3.4 Project Structure
3.5 Routing
3.6 Middleware (Introduction)
3.7 REST APIs
3.8 CRUD APIs
3.9 JSON
3.10 API Testing using Postman

Unit-4. Database Integration & Full Stack Application

4.1 Introduction to MongoDB
4.2 Collections & Documents
4.3 CRUD Operations
4.4 Connecting MongoDB with Node.js
4.5 Frontend-Backend Integration
4.6 GitHub Repository Management
4.7 Deployment Overview
4.8 Career Roadmap for Full Stack Developer

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	-	-	1	2	-	1	-	-	1.00	3	2	1	1	1.75
CO-2	3	2	3	1	-	-	2	3	-	2	-	-	1.33	3	3	2	1	2.25
CO-3	3	3	3	2	1	-	2	3	-	2	1	-	1.67	3	3	3	2	2.75
CO-4	3	3	2	2	2	1	2	3	1	2	1	1	1.92	3	3	3	2	2.75
Avg	3	2.5	2.5	1.5	0.75	0.25	1.75	2.75	0.25	1.75	0.50	0.25	1.52	3	2.75	2.25	1.5	2.38

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Interactive lectures with demonstrations of web technologies and architecture.
CO-2 (Unit: 2)	• Live coding sessions with guided hands-on practice using React.js.
CO-3 (Unit: 3)	• Demonstration-based teaching of Node.js, Express.js, REST APIs, and MongoDB concepts.
CO-4 (Unit: 4)	• Studio-based learning through case studies, application integration, and project discussions

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Presentation
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	12	8
	CO-2	10	5		
	CO-3	10	5		
	CO-4	10	5		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

References

Books:

- Duckett, J. (2022). HTML and CSS: Design and Build Websites (Revised Edition). Wiley.
- Duckett, J. (2024). JavaScript and jQuery: Interactive Front-End Web Development (2nd Edition). Wiley.

- Banks, A., & Porcello, E. (2024). Learning React: Modern Patterns for Developing React Apps (3rd Edition). O'Reilly Media.
- Haverbeke, M. (2024). Eloquent JavaScript (4th Edition). No Starch Press.
- Brown, E. (2019). Web Development with Node and Express (2nd Edition). O'Reilly Media.
- Holowaychuk, T. (2019). Express in Action. Manning Publications.
- Chodorow, K. (2019). MongoDB: The Definitive Guide (3rd Edition). O'Reilly Media.
- Freeman, E., Robson, E., Sierra, K., & Bates, B. (2021). Head First JavaScript Programming (2nd Edition). O'Reilly Media.

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010345035	Name of Course Advance Database Management System	Compulsory
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: **Apply** advanced SQL programming and concurrency control mechanisms to ensure data integrity, automate server-side logic, and manage multi-user transaction environments effectively.

CO-2: **Design** and optimize relational database schemas by applying Enhanced Entity-Relationship (EER) modeling and normalization techniques up to 5NF to eliminate data redundancy.

CO-3: **Analyze** the architectures of Distributed, Parallel, and NoSQL databases, evaluating trade-offs between consistency and availability (CAP Theorem) for large-scale data systems.

CO-4: **Evaluate** and implement modern database solutions for AI, utilizing vector databases and autonomous tuning structures to support Large Language Models (LLMs) and high-dimensional data search.

Detailed Syllabus

Unit-1. Transaction Management & SQL Programming

1.1 **Concurrency Control:** Locking and Isolation, Deadlock detection and prevention, Serializability (Conflict and View), Granularity of Locking, Intent Locking.

1.2 **Isolation Levels:** Read Uncommitted, Read Committed, Repeatable Read, and Serializable.

1.3 **SQL Mastery:** * **Management:** Advanced Database and Table creation (Constraints: Check, Unique, Foreign Key).

1.3.1 **Manipulation:** Complex Select, Insert, Delete, and Update logic.

1.4 **Automation:** Triggers (Row-level vs. Statement-level), Stored Procedures, and Functions.

Unit-2. Database Design & Normalization

2.1. **Modeling:** Advanced ER Diagrams, EER (Specialization, Generalization, Aggregation).

2.2. **Functional Dependencies:** Non-loss Decomposition and Dependency Preservation.

2.3. **Normal Forms:** 1NF, 2NF, 3NF, BCNF, 4NF, and 5NF.

2.4. **Design Theory:** Multi-valued dependencies and Join dependencies.

Unit-3. Distributed and Parallel Databases, NoSQL

3.1. **Distributed DBMS (DDBMS):** Architecture, Fragmentation (Horizontal/Vertical), Replication, Data Localization.

3.2. **Distributed Transactions:** Concurrency Problems (The Three Concurrency Problems Revisited), Two-Phase Commit (2PC) and Three-Phase Commit (3PC) protocols.

3.3. **Parallel Databases:** I/O Parallelism, Inter-query vs. Intra-query parallelism, Parallel Sort/Join algorithms.

3.4. **Theory:** CAP Theorem and SQL Support for Distributed Environments.

3.5. **NoSQL Categories:** Key-Value stores (Redis), Column-family (Cassandra), Document-oriented (MongoDB), Graph databases (Neo4j).

3.6. **Consistency Models:** ACID vs. BASE (Basically Available, Soft state, Eventual consistency).

3.7. **Indexing:** R-Trees, Grid files, and Multidimensional indexes for spatial and high-dimensional data.

Unit-4. Database Systems for AI & LLMs (New Component)

4.1. **Vector Databases:** Concepts of Embedding, Vector Search, and Similarity Measures (Cosine, Euclidean, Dot Product).

4.2. **Autonomous Databases:** AI-driven Query Tuning (Self-tuning), Learned Index Structures, and AI-based anomaly detection for security.

4.3. **Modern Tools:** Introduction to ChromaDB, Pinecone, or Milvus for AI applications.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	3	3	1	1	2	1	3	2	2	1	1	1.92	3	3	1	1	2
CO-2	3	3	3	1	–	1	1	2	1	2	1	–	1.5	3	2	1	1	1.75
CO-3	3	3	3	2	1	2	1	3	2	2	2	1	2.08	3	3	2	1	2.25
CO-4	3	3	3	2	1	2	1	3	2	3	2	1	2.17	3	3	2	2	2.5
Avg	**3**	**3**	**3**	**1.5**	**0.75**	**1.75**	**1**	**2.75**	**1.75**	**2.25**	**1.5**	**0.75**		**3**	**2.75**	**1.5**	**1.25**	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Classroom Teaching, Tutorial, Presentation, Seminar

Assessment Method

Continuous Comprehensive Evaluation 40 Marks	**COs**	**Marks**	**Exam Component**		
			Class Test	**Assignment**	**Presentation**
	CO-1	10	5	5	-
	CO-2	10	5	-	5
	CO-3	10	5	2.5	2.5
	CO-4	10	5	2.5	2.5
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

References

Books:

- Database System Concepts by Silberschatz, Korth, and Sudarshan.
- Principles of Distributed Database Systems by M. Tamer Özsu.
- Database Management Systems by Raghu Ramakrishnan and Johannes Gehrke.
- Vector Databases for Beginners (Technical Documentation from Milvus/Pinecone).

Program – MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010245036	Name of Course Lab based on Object Oriented Programming Using Java	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Develop Java programs implementing fundamental programming constructs and object-oriented principles such as classes, objects, constructors, inheritance, polymorphism, abstraction, and interfaces.

CO-2: Create modular Java applications using packages and implement exception handling mechanisms, including custom exceptions, to improve program reliability and maintainability.

CO-3: Implement file processing and multithreaded applications using Java streams, file handling classes, serialization techniques, thread management, synchronization, and communication methods.

CO-4: Build efficient and reusable applications using Java Collection Framework components and Generics for data storage, manipulation, traversal, and type-safe programming.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	1	1	2	2	2	1	1	1	1.67	3	2	1	1	1.75
CO-2	3	2	3	1	1	2	2	2	2	1	1	1	1.75	3	2	1	2	2
CO-3	3	3	3	1	1	2	2	3	3	2	1	1	2.08	3	3	2	2	2.5
CO-4	3	2	3	1	1	1	2	3	2	2	1	1	1.83	3	2	1	1	1.75
Avg	3	2.25	3	1	1	1.5	2	2.5	2.25	1.5	1	1		3	2.25	1.25	1.5	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Experimental Learning • Demonstration and Guided Practice • Project-Based Learning • Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010245037	Name of Course Lab based on Data Structures	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Teaching Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Develop, implement, and analyze linear data structures like arrays, stacks, queues, and linked lists with their applications.

CO-2: Construct and manipulate non-linear data structures like binary search trees and understand balancing mechanisms of advanced trees.

CO-3: Apply graph theory and multi-linked structures to solve computational problems and implement dynamic memory management techniques.

CO-4: Implement various searching and sorting algorithms, analyze their performance, and apply hashing techniques for efficient data retrieval.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	–	–	1	1	2	1	2	–	–	1.73	3	2	1	2	2
CO-2	3	2	3	–	–	1	–	2	1	2	–	–	1.64	3	3	1	2	2.25
CO-3	3	3	3	–	–	2	–	3	2	3	1	–	2.36	3	3	2	1	2.25
CO-4	3	2	3	–	–	2	–	2	2	2	–	–	2	3	2	2	3	2.5
Avg	3	2.25	3.0	–	–	1.5	0.25	2.25	1.5	2.25	0.25	–	1.93	3	2.5	1.5	2	2.25

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Experimental Learning • Demonstration and Guided Practice • Project-Based Learning • Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010245038	Name of Course Lab based on Full Stack Web Development	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO1 Develop responsive web pages using HTML5, CSS3, JavaScript, and React.js.

CO2 Implement reusable React components, routing, forms, state management, and API integration

CO3 Develop and test RESTful backend services using Node.js, Express.js, and MongoDB..

CO4 Design, integrate, debug, test, and demonstrate a complete full-stack web application using GitHub and modern development tools

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	1	2	3	1	2	1	1	1.67	3	2	1	2	2
CO-2	3	3	3	1	–	1	2	3	2	2	2	1	1.92	3	3	1	2	2.25
CO-3	3	3	3	2	1	2	2	3	2	2	2	1	2.17	3	3	2	2	2.5
CO-4	3	3	3	2	1	3	2	3	3	3	2	2	2.5	3	3	2	3	2.75
Avg	3	2.75	3	1.5	0.5	1.75	2	3	2	2.25	1.75	1.25		3	2.75	1.5	2.25	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Hands-on laboratory exercises for developing responsive web pages and React applications. • Guided programming sessions for implementing React components, routing, and API integration • Practical implementation and testing of backend applications using Node.js, Express.js, and MongoDB • Project-based learning through full-stack application development, debugging, and project demonstration

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 1

Course Code 266010245039	Name of Course Lab based on Advance Database Management System	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Design and create relational databases using SQL Data Definition Language (DDL) commands and implement integrity constraints, keys, and relationships.

CO-2: Manipulate and retrieve data using SQL Data Manipulation Language (DML) and query statements, and develop database triggers for automation of database operations.

CO-3: Analyze and normalize database schemas using ER modeling, functional dependencies, and normalization techniques up to BCNF to reduce redundancy and maintain data integrity.

CO-4: Implement transaction management, locking mechanisms, isolation levels, and concurrency control techniques to ensure consistency and reliability in multi-user database environments.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	1	3	1	-	2	1	2	1	2	1	-	1.42	3	2	1	1	1.75
CO-2	3	2	3	1	-	2	1	2	1	2	1	-	1.50	3	2	1	1	1.75
CO-3	3	3	3	2	-	2	1	2	1	2	1	-	1.67	3	2	2	1	2.00
CO-4	3	2	3	1	1	3	1	2	2	2	1	1	1.83	3	3	2	1	2.25
Avg	3.00	2.00	3.00	1.25	0.25	2.25	1	2	1.25	2	1.00	0.25		3	2.25	1.50	1	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Experimental Learning, Demonstration and Guided Practice • Project-Based Learning, Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010345009	Name of Course Operating System	Elective
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the fundamental concepts of operating systems, including system architecture, kernel functions, and the internal structure of files and directories.

CO-2: Describe and utilize file system calls and understand process states, transitions, and memory layout in the operating system.

CO-3: Manage processes by implementing context switching, process creation, signal handling, and process termination.

CO-4: Apply process scheduling algorithms and understand memory management concepts along with system boot procedures and initialization processes.

Detailed Syllabus

Unit-1.

1.1 **Introduction to Operating System and File Structure:** General overview of the system, System Structure, User Perspective, Operating System Service, Assumptions about Hardware, Introduction to the kernel, Architecture of the Operating System, Introduction to the system concepts, Kernel Data Structures, System Administration, Internal Representation of Files, Structure of a Regular File, Directories, Conversion of a path name, Super Block, Creation of a new file, Allocation of Disk Blocks, Other File Types

Unit-2.

2.5 **File System Calls and Process Sub System:** System Calls for the File System (without algorithm) - Open, Read, Write, File and Record Locking, Close, File Creation, Change Directory and Change Root, Change Owner and Changing Mode, The structure process, Process states and transitions, Layout of system memory

Unit-3.

3.1 **Process functions:** The context of a process, Saving the context of a process, Manipulation of the process address space, Sleep

3.2 **Control of the process:** Process Control, Process Creation, Signals, Process termination, Awaiting process termination, invoking other programs, the user ID of a process, Changing the size of a process

Unit-4.

4.1 Process Scheduling and Memory Management: Process scheduling and time, Process scheduling, System calls for time, Memory Communication, The shell, System Boot, The INIT process

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	1	1	1	2	1	2	1	1	1.58	3	2	1	1	1.75
CO-2	3	2	3	1	1	1	1	2	1	2	1	1	1.67	3	2	1	1	1.75
CO-3	3	3	3	1	1	1	1	2	2	2	1	1	1.83	3	2	2	1	2
CO-4	3	3	3	2	2	1	1	2	2	2	2	2	2.08	3	3	2	2	2.5
Avg	3	2.5	2.75	1.25	1.25	1	1	2	1.5	2	1.25	1.25	1.79	3	2.25	1.5	1.25	2.00

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy	
CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Classroom Teaching, Group Discussion, Assignment, Examples, Questioning

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Morris Bache, The design of Unix Operating system, PHI
- Meeta Gandhi, Tilak Shetty, Rajiv Shah,The C Odysse, BPB
- Abraham Silberschatz , Peter B. Galvin, et al, Operating System Concepts, Wiley, 2019
- Yashavant Kanetkar, Unix Shell Programming by, BPB publications, 2003

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central), e-PGPathsala(e-PGPathshala (inflibnet.ac.in))

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010345010	Name of Course Mobile Application Development	Elective
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand the fundamental components of Android applications, including activities, services, and intents, and configure the AndroidManifest file to manage application behavior and permissions effectively.

CO-2: Organize and utilize Android application resources efficiently and design intuitive user interfaces using layout managers, screen elements, and animations.

CO-3: Integrate Android data storage techniques such as SQLite and content providers, and implement RESTful web services to enable seamless data exchange within and across mobile applications.

CO-4: Apply advanced Android APIs related to networking, telephony, location (Google Maps), and hardware sensors to develop feature-rich, context-aware mobile applications.

Detailed Syllabus

Unit-1.

1.1 **Android Application Design Essentials:** Anatomy of an Android applications, Application Context, Activities, Services, Intents

1.2 **Android Application Design Essentials:** Receiving and Broadcasting Intents, Android Manifest File and its common settings.

Unit-2.

2.1 **Manifest File:** Using Intent Filter, Permissions, Managing Application resources in a hierarchy, Working with different types of resources**.**

2.2 **Android User Interface Design Essentials:** User Interface Screen elements, Designing User Interfaces with Layouts, Drawing and Working with Animation

Unit-3.

3.1 **Using Common Android APIs:** Using Android Data and Storage APIs, Managing data using SQLite, Webservice (SOAP and REST), REST Webservice creation and utilization of webservice in Android Application, Sharing Data between Applications with Content Providers.

Unit-4.

4.1 Using Common Android APIs: Android Networking APIs , Android Web APIs, Android Telephony APIs, Google MAP in Android application, Accessing Android's Hardware Sensors (orientation sensors, light sensors)

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	2	–	2	1	2	2	2	2	2	2.09	3	2	2	12	2.25
CO-2	3	1	3	3	2	2	2	2	1	2	2	2	2.08	3	3	2	2	2.5
CO-3	3	3	3	2	2	2	1	3	2	2	2	2	2.25	3	3	2	2	2.5
CO-4	3	2	3	2	3	2	1	3	2	2	2	2	2.25	3	3	2	2	2.5
Avg	3	2	3	2.25	2.33	2	1.25	2.50	1.75	2	2	2	2.17	3	2.75	2	2	2.44

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy	
CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Classroom Teaching, Tutorial, Presentation, Seminar

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Presentation
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	5	-
	CO-2	10	5	-	5
	CO-3	10	5	2.5	2.5
	CO-4	10	5	2.5	2.5
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

References

Books:

- Android Wireless Application Development by LaurenDarcey and Shane Conder, 3rd Edition, Pearson Education.
- Professional Android 2 Application Development by Reto Meier, Wiley India Pvt Ltd, 2011.
- Beginning Android by Mark L Murphy, Wiley India Pvt Ltd, 2009.
- Pro Android by Sayed Y Hashimi and Satya Komatineni, Wiley India Pvt Ltd, 2009.

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010345011	Name of Course Web Technology (Lab based)	Compulsory
Credit: 03	Teaching Scheme: Theory (0) - Practical (90)	Teaching Hours: 90

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Develop dynamic web pages using DHTML, JavaScript, and DOM. And describe and distinguish the architecture of client-side and server-side web applications. Establish database connectivity using JDBC to create data-driven applications.

CO-2: Build Java web applications using Servlets that handle HTTP requests, manage sessions through hidden form fields, URL rewriting, cookies and session API and interact with databases using JDBC for persistent data processing.

CO-3: Design interactive and responsive web applications using Java Server Pages (JSP), leveraging JSP scripting elements, session tracking, and JavaBeans for seamless data exchange and form handling.

CO-4: Implement Java web applications using the MVC design pattern and create well-structured XML documents with DTD and XML Schema for effective data representation and validation.

Detailed Syllabus

Unit-1. Data manipulation and Store data, Web Concepts, DHTML and Java Editions

1.1 **Database Handling using JDBC:** Java Database Connectivity, Driver, Driver Types, Compare and advantages of drivers, DriverManager, Connection, Statement, Prepared Statement, Callable Statement, ResultSet, Result set metadata. Overview of the Internet, Web as a platform and its components. Form processing at the client side. DHTML and its components. Dynamic page using DOM, CSS, and Java Script.

1.2 **Introduction to Request:** Response Architecture, Web application and HTTP Protocol, Tomcat application server and its structure, Java Web Application Architecture, Understanding HTTP Status Codes, HTTP Request ad Response Headers, Overview of Java Editions.

1.3 **Tomcat – The Servlet-JSP Container/Engine:** Introduction, Installation, and configuration, Understating the responsibility of Container/Engine. Create and deploy web application.

Unit-2. Servlet API and Session Management

2.1 **Servlet Model:** Servlet: What and why? Servlet Life Cycle, HTTP Methods Structure and Deployment descriptor, Comparison with existing technologies, Servlet Interface, Servlet Context and Servlet Config interface, Generic Servlet, Http Servlet, Steps to create a Java web application in Tomcat, Handling Client Request- Reading Request Headers, reading request data in Servlet and Generate dynamic content/response. Request Redirection and Dispatching, Servlet- catch form data sent from client, process it, and store it on database. JDBC (Java Database Connectivity) and how it can be used within servlet.

2.2 **Session Tracking and Management:** Session Tracking: What and Why? Understanding Session Timeout and Session Tracking - Hidden Form Field, URL Rewriting, Cookies, HTTP Session

2.3 **Cookies:** Create Cookie, remember user data, Deleting Cookies, Sending and Receiving Cookies, Differentiating Session Cookies from Persistent Cookies, Using Cookies to Remember User Preferences.

Unit-3.

3.1 **Session API:** Session API, Session Tracking: session creation, session management, session timeout, accumulating and retrieve User Data, Encoding URLs.

3.2 **Java Server Pages (JSP):** Compare Servlet and JSP, Overview of JSP, Advantages of JSP, JSP Comment, Life Cycle of JSP page, JSP API, JSP Expression, JSP Scriptlet, JSP Declaration, JSP Directives, JSP Standard Action, JSP implicit Objects, JSP Directive, JSP Scripting elements, JSP Action Elements:

jsp:forward, jsp:include, jsp:useBean, jsp:setProperty & jsp:getProperty, Java Bean and JSP Communication, Exception Handling, JSP Session and Cookies Handling, JSP Session Tracking, JSP-catch form data sent from client, process it, and store it on database.

Unit-4.

4.1 Model-View-Controller (MVC) Application Design with Servlet-JSP: Introduction, MVC pattern Layer: Model, View, and Controller. Role of Servlet and JSP in MVC. Role and responsibility of Model, View, and Controller in MVC, Advantages of MVC. Role of Servlet, JSP, Java bean, and POJO in MVC.

4.2 XML Technology

4.2.1 Extensible Markup Language (XML): Introduction and Overview, XML, Understanding the purpose and difference of HTML and XML, History and application of XML, XML Syntax, XML Document Structure and Building Blocks of XML Documents, XML Parsers, Well-formed and valid XML Documents, XML Namespace, Understating DOM, Types of Elements

4.2.2 Document Type Definition (DTD): Introduction to DTD, Purpose of DTD, Create Internal and External DTD, referencing a DTD in an XML Document, defining building blocks of XML documents - Elements, Attributes, Entities, PCDATA, CTADA, Declaring Elements, Attributes and Entity.

4.3 XML Schema: Purpose of XML Schema, Advantages of XML Schema, Comparison with DTD, Understanding Why XML Schema is better than DTD, Create XML Schema Document (.XSD), Referencing a Schema in an XML Document, defining building blocks of XML documents using Schema, XML Schema Date Types, Understating use of Restriction, Occurrence, and Indicators with examples.

Mapping Matrix of POs, PSOs, and Cos

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	2	2	3	2	2	2	1	2.1	3	2	2	2	2.25
CO-2	3	3	3	1	–	3	2	3	2	2	2	1	2.27	3	3	2	2	2.5
CO-3	3	2	3	1	–	3	2	3	2	2	2	1	2.18	3	3	2	2	2.5
CO-4	3	3	3	2	1	3	2	3	2	2	2	2	2.33	3	3	2	3	2.75
Avg	3	2.5	3	1.25	1	2.75	2	3	2	2	2	1.25	2.22	3	2.75	2	2.25	2.5

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Experiential Learning • Demonstration–Practice–Reflection: Demonstration of concepts and tools, Hands-on Practice/Experiments, Debugging, Discussion, and Reflection • Collaborative Learning, Outcome-Based Learning, Questioning and Quiz, Continuous Evaluation, Project-Based Learning

Assessment Method

Continuous Comprehensive Evaluation 40 Marks	**COs**	**Marks**	**Exam Component**			
			Quiz	**Practical Test**	**Assignment**	**Participation/ Performance**
	CO-1	10	2.5	2.5	13	7
	CO-2	10	2.5	2.5		
	CO-3	10	2.5	2.5		
	CO-4	10	2.5	2.5		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- "Java Servlet Programming", by Jason Hunter, William Crawford, O'Reily Publication
- "Head First Servlets and JSP" by Bryan Basham, Kathy Sierra, Bert Bates, O'Reily Publication
- "Professional XML", by Mark Birbeck, Wrox Publication
- "Core Servlets and Java Server Pages" Volume – 2", Pearson Education
- "Java Server Programming", A Press Publication
- "Pro JSP 2" by Simon Brown, Sam Dalton, Daniel Jepp, David Johnson, Sing Li, and Matt Raible, Apress Publication
- "Web Technologies Black Book", Dreamtech Press, Edition 2010
- "Web Enabled Commercial Application Development Using HTML, DHTML, PERL, Java Script", by Ivan Bayross, BPB Publications, Revised Edition

Online Resources & Tools:

- **Apache Tomcat** http://tomcat.apache.org

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010445012	Name of Course Computer Network	Compulsory
Credit: 04	Teaching Scheme: Theory (60) - Practical (0)	Teaching Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the structure and purpose of layered network models (OSI and TCP/IP), describe physical transmission media, and apply basic error detection and correction techniques used in the data link layer.

CO-2: Analyze and implement data link control mechanisms and evaluate various routing algorithms and addressing schemes in both IPv4 and IPv6.

CO-3: Describe the responsibilities of the transport layer and demonstrate how reliable communication, congestion control, and connection management are achieved in computer networks.

CO-4: Examine key application layer protocols (such as DNS, SMTP, POP3) and apply foundational concepts of network security, including cryptography, digital signatures, and authentication protocols.

Detailed Syllabus

Unit-1.

1.1 **Introduction:** Need of Data Communication and Applications, Network Models, TCP/IP and OSI Layering Models

1.2 **Physical Layer:** Transmission Media, Wired and Wireless Physical Layer

1.3 **Data Link Layer – Error Detection and Correction:** Introduction and Duties of Data Link Layer, Types of Errors, Redundancy, Detection Versus Correction, Forward Error Correction Versus Retransmission, Error Detection, Error Correction, Block Coding, Linear Block Codes, Cyclic Codes

Unit-2.

2.1 **Data Link Layer – Data Link Control:** Data Link Control and Protocols, Flow and Error Control and its mechanism, Noiseless and Noisy Channels, Bluetooth, Architecture, Applications, Profiles, Pairing Process

2.2 **Network Layer:** Introduction, Duties of Network Layer, Routing, Accounting, Global Machine Level addressing, Connection Oriented and Connectionless Forwarding, Forwarding Examples, Routing Algorithms, Distance Vector Routing, Link State Routing, Border Gateway Protocol, Congestion and its Control, IPv4 Addresses, Address Space, Notations, Classful Addressing, Classless Addressing, Subnetting and Supernetting, IPv6 Addresses, Structure, Address Space, ICMP

Unit-3.

2.1 Transport Layer: Introduction, Duties of Transport Layer, Multiplexing, Demultiplexing and Port Numbers, Service to other Layers, Transport Layer of the Internet, Process Level Addressing, End to End Solutions, Connection Management at the Transport Layer, Delayed Duplicates, Connection Establishment, Connection Release, Congestion Control, Detecting Congestion, Reacting to Congestion, Fast Recovery, Flow Control, Communication Primitives

Unit-4.

4.1 Application Layer: Introduction, Domain Name System, Domain Name Space, Registration Process, Name Servers, Resource Records, Mailing System, SMTP, POP3 and IMAP, Webmail, SNMP, Network Protocol Analyzer, Wireshark and its Applications and Features

4.2 Information Security: Introduction to Network Security, Cryptography, Digital Signatures, Public Key Management, Authentication Protocol, Authentication based on Shared Secret Key, Information Security

Mapping Matrix of POs, PSOs, and Cos

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	1	2	-	1	2	1	2	-	1	-	1	1.55	3	2	1	2	2
CO-2	3	2	3	1	1	1	1	2	2	1	-	1	1.63	3	3	1	2	2.25
CO-3	3	1	3	-	-	1	1	2	1	1	-	-	1.62	3	3	2	1	2.25
CO-4	3	1	2	1	2	3	1	3	1	1	-	2	1.81	3	2	2	3	2.5
Avg	3	1.25	2.5	1	1.33	1.75	1	2.25	1.33	1	-	1.33	1.65	3	2.5	1.5	2	2.25

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Lecture, Assignment, Demonstration, Example

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Computer Network, AndrewS.Tanenbaum
- Computer Network, Bhushan Trivedi
- Introduction to Data Communication and Networking,BehrouzForouzan
- Computer Network, Natalia Olifer, VictorOlifer
- Data and Computer Communication, William Stallings

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central)
- e-PGPathsala(e-PGPathshala (inflibnet.ac.in))
- NPTEL

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010345013	Name of Course Software Design Pattern (GOF)	Compulsory
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Analyze and model real-world systems using object-oriented principles, applying concepts such as classes, associations, generalization, inheritance, and aggregation to develop effective class diagrams.

CO-2: Identify and apply appropriate creational design patterns such as Factory Method, Abstract Factory, Singleton, and Prototype to solve recurring object creation problems in software design.

CO-3: Implement structural and behavioral design patterns, including Adapter, Decorator, Proxy, Strategy, and Observer, to enhance software flexibility and maintainability.

CO-4: Evaluate and integrate design patterns and anti-patterns within modern development frameworks (e.g., .NET, Java) through real-world case studies and practical applications.

Detailed Syllabus

Unit-1.

1.1 **Object Modeling:** Introduction to key principles of Object-oriented design, Object Orientation Analysis and Design, Object Oriented Development and Themes, Importance of Modelling, principles of modelling, Objects, Classes, Class Diagrams, Values and Attributes, Operations and Methods, Link and Association concepts -Links and Associations, Multiplicity, Association and Names, Ordering, Association Classes, Qualified Association, Generalization and Inheritance, Aggregation ,Abstract classes, Generalization as extension and Restriction, Grouping Constructs, Sample Object modelling

Unit-2.

2.1 **Design Patterns (GOF):** Introduction to design Pattern, Describing design Patterns, The catalog of Design Patterns, selecting design pattern and solve design problems

2.2 **Creational Patterns:** Abstract factory, Factory Method, Singleton, Prototype

Unit-3.

3.1 **Structural Patterns:** Adapter, Decorator, Façade, Proxy

3.2 **Behavioral Patterns:** Chain of Responsibility Pattern, State, Strategy, Observer

Unit-4.

4.1 Advanced Topics and Case Studies: Combining pattern, anti-pattern, Case Study of and Pattern in modern frameworks like .net , java etc.

Mapping Matrix of POs, PSOs, and COs

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	1	1	1	2	–	2	–	–	**2**	3	2	1	2	**2**
CO-2	3	2	3	–	–	2	–	3	–	2	1	–	**2.28**	3	3	2	2	**2.5**
CO-3	3	2	3	1	1	2	–	3	2	2	–	–	**2.42**	3	3	2	2	**2.5**
CO-4	3	3	2	1	1	2	1	3	2	3	2	1	**2.09**	3	2	2	3	**2.5**
Avg	**3**	**2.25**	**2.75**	**1**	**1.5**	**1.75**	**1**	**2.75**	**2**	**2.25**	**1.5**	**1**	**2.20**	**3**	**2.5**	**1.75**	**2.25**	**2.38**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	● Problem Solving, Examples, Questioning, Assignment
CO-2 (Unit: 2)	● Problem Solving, Examples, Questioning, Assignment, Experimental
CO-3 (Unit: 3)	● Problem Solving, Examples, Questioning, Assignment, Experimental
CO-4 (Unit: 4)	● Problem Solving, Examples, Questioning, Assignment, Experimental, Demonstration, Project

Assessment Method

Continuous Comprehensive Evaluation 40 Marks	**COs**	**Marks**	**Exam Component**				
			Class Test	**Assignment**	**Presentation**	**Case Study**	**Involvement**
	CO-1	10	5	5	4	5	6
	CO-2	10	5				
	CO-3	10	5				
	CO-4	10	5				
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**				
	CO-1	15	Term End Examination				
	CO-2	15					
	CO-3	15					
	CO-4	15					

References

Books:

- Object – Oriented Modeling and Design by James Rumbaugh, Michael Blaha
- Design Patterns Elements of Reusable Object-Oriented Software by Erich Gama, Richard Helm, Ralph Johnson, John Vlissides, Pearson Education
- Head First Object –Oriented Analysis & Design by Brett D. McLaughlin, Gary Pollice & David West, O'REiLLY
- Head First Design Pattern by Eric Freeman & Elisabeth Freeman, O'REiLLY

Online Resources & Tools:

- https://www.edulib.in//userLib/subjectTopics/553

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010345014	Name of Course Software Engineering	Compulsory
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the fundamental concepts of software engineering, including various software process models and agile methodologies, to understand software development life cycles.

CO-2: Analyze and document software requirements effectively using techniques such as requirement elicitation and UML modeling to create comprehensive Software Requirement Specifications (SRS).

CO-3: Apply design principles and testing strategies to develop robust software architectures and validate software quality through systematic testing approaches.

CO-4: Manage software projects by employing project planning, risk management, software metrics, and quality assurance practices to ensure timely and quality software delivery.

Detailed Syllabus

Unit-1.

1.1 **Introduction to Software and Software Engineering:** The Evolving Role of Software, Software Engineering: A Layered Technology, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Component-Based Development, Agility and Agile Process model, Extreme Programming

Unit-2.

2.1 **Requirement Analysis and Specification:** Understanding the Requirement, Requirement Modelling, Requirement Specification (SRS), Requirement Analysis and Requirement Elicitation, Requirement Engineering

2.2 **Introduction to UML:** UML Building Blocks, Modelling Views, Introduction to Use Case, Use Case Diagrams, State Diagrams, Sequence Diagrams, Activity Diagrams, Component Diagrams, Packages and Foundation

Unit-3.

3.1 Software Design: Design Concepts and Design Principal, Architectural Design, Component Level Design, User Interface Design, Web Application Design

3.2 Software Testing: Testing Strategies, Testing Techniques, Test Cases, Testing Conventional Applications, Testing Object Oriented Applications

Unit-4.

4.1 Software Project management: Software Metrics : Process, Product and Project Metrics, Software Project Estimations, Software Project Planning, Project Scheduling & Tracking, Risk Analysis & Management

4.2 Software Quality management: Quality Concepts and Software Quality Assurance, Software Reviews (Formal Technical Reviews), Software Reliability, The Quality Standards : ISO 9000, CMM, Six Sigma for SE, SQA Plan

Mapping Matrix of POs, PSOs, and COs

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	–	2	–	–	2	–	2	–	1	–	–	**2**	3	2	1	2	**2**
CO-2	2	2	3	1	–	2	–	2	1	–	–	–	**1.88**	3	3	1	2	**2.25**
CO-3	3	–	3	–	2	2	–	3	1	–	–	–	**2.29**	3	3	2	1	**2.25**
CO-4	2	3	2	1	–	3	2	2	3	2	3	1	**2.18**	3	3	2	3	**2.75**
Avg	**2.50**	**2.33**	**2.50**	**1**	**2**	**2.25**	**2**	**2.25**	**1.67**	**1.50**	**3**	**1**	2.09	**3**	**2.27**	**1.75**	**2**	2.31

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	● Lecture, Seminar, Assignment, Case-Study, Examples
CO-2 (Unit: 2)	
CO-3 (Unit: 3)	
CO-4 (Unit: 4)	

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Presentation
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	7	13
	CO-2	10	5		
	CO-3	10	5		
	CO-4	10	5		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

References

Books:

- Software Engineering – A Practitioner's Approach by Roger S. Pressman, McGraw-Hill International Edition, 7th Edition.
- Software Engineering by Ian Sommarville, Printice_Hall India.
- Software Engineering by Pankaj Jalote, Narosa Publication.

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central)
- e-PGPathsala (e-PGPathshala (inflibnet.ac.in))

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010245015	Name of Course Lab based on Operating System	Elective-I
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Develop and execute shell scripts for automation, file management, and demonstrate process-related commands and system programming in UNIX/Linux.

CO-2: Develop File sharing and resource sharing programs using low level system calls.

CO-3: Implement and analyze process creation, management of process using pipe and inter process communications using signals in process sub systems.

CO-4: Implement message queue and semaphore for process management and inter process communications.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	1	1	1	2	1	2	1	1	**1.58**	3	2	1	1	**1.67**
CO-2	3	2	3	1	1	1	1	2	1	2	1	1	**1.67**	3	2	1	1	**1.75**
CO-3	3	3	3	1	1	1	1	2	2	2	1	1	**1.83**	3	2	2	1	**1.92**
CO-4	3	3	3	2	2	1	1	2	2	2	2	2	**2.08**	3	3	2	2	**2.17**
Avg	**3**	**2.5**	**2.75**	**1.25**	**1.25**	**1**	**1**	**2**	**1.5**	**2**	**1.25**	**1.25**	**1.79**	**3**	**2.25**	**1.5**	**1.25**	**1.88**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Experimental Learning, Demonstration and Guided Practice • Project-Based Learning, Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Participation/ Performance
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	13	7
	CO-2	10	5		
	CO-3	10	5		
	CO-4	10	5		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010245016	Name of Course Lab based on Mobile Application Development	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand the structure and components of Android applications and apply intent-based communication between components.

CO-2: Design effective and user-friendly interfaces using Android UI components, resources, layouts, and animation techniques.

CO-3: Utilise Android APIs for local storage and integrate web services to handle data-driven mobile applications.

CO-4: Implement Android networking, telephony, and sensor APIs.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	2	2	3	2	2	1	1	**2**	3	3	2	2	**2.5**
CO-2	3	1	3	2	1	2	2	3	1	1	1	1	**1.75**	3	2	2	1	**2**
CO-3	3	2	3	2	–	2	1	3	2	2	2	1	**2.09**	3	3	2	2	**2.5**
CO-4	3	1	3	2	1	2	1	3	2	–	–	–	**2**	3	2	2	1	**2**
Avg	**3**	**1.50**	**3**	**1.75**	**1**	**2**	**1.50**	**3**	**1.75**	**1.67**	**1.33**	**1**	**1.96**	**3**	**2.5**	**2**	**1.5**	**2.25**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Experimental Learning • Demonstration and Guided Practice • Project-Based Learning • Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Participation/ Performance
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	13	7
	CO-2	10	5		
	CO-3	10	5		
	CO-4	10	5		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 2

Course Code 256010245017	Name of Course Lab based on Software Design Pattern (GOF)	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Teaching Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Apply the principles of object-oriented design and develop object models using class diagrams, associations, generalization, aggregation, and other UML constructs for real-world systems.

CO-2: Design and implement solutions to software creation problems using appropriate creational patterns such as Singleton, Factory Method, Abstract Factory, and Prototype.

CO-3: Analyze software design problems and apply structural and behavioral patterns, including Adapter, Decorator, Proxy, Facade, Strategy, Observer, Chain of Responsibility, and State.

CO-4: Integrate multiple design patterns effectively in large-scale systems, identify and refactor anti-patterns, and analyse the use of patterns in modern frameworks and case studies.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	2	1	1	1	2	2	2	1	1	1.75	3	2	2	2	2.25
CO-2	3	2	3	1	1	1	1	2	2	2	1	1	1.67	3	2	2	1	2
CO-3	3	2	3	2	1	2	1	2	2	2	1	1	1.83	3	3	2	2	2.5
CO-4	3	3	3	2	1	2	2	3	3	2	2	2	2.33	3	3	3	2	2.75
Avg	3	2.25	3	1.75	1	1.5	1.25	2.25	2.25	2	1.25	1.25	1.90	3	2.5	2.25	1.75	2.38

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Experimental Learning • Demonstration and Guided Practice • Project-Based Learning • Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Class Test	Assignment	Participation/ Performance
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	13	7
	CO-2	10	5		
	CO-3	10	5		
	CO-4	10	5		
Term-End Evaluation 60 Marks	COs	Marks	Exam Component		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510245040	Name of Course Microservices Using Spring Boot	Compulsory
Credit: 02	Teaching Scheme: Theory (30) - Practical (0)	Teaching Hours: 30

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the principles of software architecture, microservices, Spring Framework, Spring Boot, and RESTful web services for developing modern enterprise applications.

CO-2: Develop RESTful web applications using Spring Boot by applying Spring Core, dependency injection, Spring MVC, and application configuration techniques.

CO-3: Design and implement database-driven applications using Spring Data JPA by performing entity mapping, repository development, CRUD operations, and transaction management.

CO-4: Develop secure, tested, and deployable Spring Boot applications by applying Spring Security, software testing techniques, and container-based deployment concepts.

Detailed Syllabus

Unit 1: Introduction to Microservices and Spring Boot

1.1 **Introduction to Software Architecture:** Introduction to Software Architecture, Monolithic Architecture, Characteristics of Monolithic Architecture, Advantages and Limitations of Monolithic Architecture, Evolution from Monolith to Microservices, Microservices Architecture, Characteristics of Microservices, Advantages and Challenges of Microservices, Monolithic vs. Microservices Architecture.

1.2 **Spring Framework and Spring Boot:** Need for Spring Framework, Features of Spring Framework, Spring Framework Architecture, Spring Ecosystem Overview, Introduction to Spring Boot, Features of Spring Boot, Spring Boot Architecture, Spring Initializr and Project Structure, Development Environment Setup.

1.3 **REST Fundamentals:** REST Architectural Principles, REST Best Practices, Resource Design, URI Design, Versioning, Idempotency.

Unit 2: Spring Boot Application Development and RESTful Web Services

2.1 Spring Core: Inversion of Control (IoC), Dependency Injection (DI), Constructor Injection vs. Field Injection, Spring Beans, Bean Scopes, Bean Lifecycle, Spring Annotations, Configuration Classes.

2.2 Spring Boot Development: Spring Boot Auto Configuration, Starter Dependencies, Configuration using application.properties, Profiles, Environment Properties, Logging, CommandLineRunner.

2.3 **Spring MVC:** Spring MVC Architecture, Controllers, Request Mapping, Request Parameters, Path Variables, Request Body and Response Body, Validation, Exception Handling.

Unit 3: Data Persistence using Spring Data JPA

3.1 **Spring Data JPA:** Introduction to Data Persistence, Object Relational Mapping (ORM), Java Persistence API (JPA), Hibernate Framework, Spring Data JPA Architecture, Database Configuration, Entity Mapping, Repository Pattern, CRUD Operations, Query Methods, DTO Pattern, Entity Relationships (One-to-One, One-to-Many, Many-to-One, Many-to-Many), Lazy vs. Eager Loading, Pagination and Sorting, Auditing, Transaction Management.

Unit 4: Spring Security, Testing and Deployment

4.1 Spring Security: Introduction to Application Security, Authentication, Authorization, Spring Security Architecture, Security Filters, Password Encoding, Role-Based Access Control, JWT Authentication (Basics), OAuth2 Basics.

4.2 Testing: Introduction to Software Testing, Unit Testing, Integration Testing, JUnit 5, Mockito, Spring Boot Testing.

4.3 **Deployment:** Packaging Spring Boot Applications, Executable JAR Files, Docker Introduction, Basic Kubernetes Concepts, Deployment Fundamentals.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	-	-	1	2	-	1	-	-	1.00	3	2	-	-	1.25
CO-2	3	2	3	-	-	1	-	3	2	1	-	-	1.25	3	3	-	-	1.50
CO-3	3	2	3	-	-	1	-	3	1	1	-	-	1.17	3	3	-	-	1.50
CO-4	3	2	3	-	1	3	1	3	2	2	-	1	1.75	3	3	1	2	2.25
Avg	3.00	2.00	2.75	0.25	0.25	1.25	0.50	2.75	1.25	1.25	0	0.25		3	2.75	0.25	1	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Interactive Lecture, Case-Based Learning, Guided Hands-on Learning, Problem-Based Learning

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Quiz
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	3	-	2
	CO-2	10	5	3	-	2
	CO-3	10	5	5	-	-
	CO-4	10	4	3	3	-
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

1. Craig Walls, *Spring Boot in Action*, Manning Publications, First Edition, 2016. *(Foundational reference; although older, it covers Spring Boot fundamentals well.)*
2. Laurentiu Spilca, *Spring Start Here: Learn What You Need and Learn It Well*, Manning Publications, Second Edition, 2024.
3. Laurentiu Spilca, *Spring Security in Action*, Manning Publications, Second Edition, 2024.
4. Mark Heckler, *Spring Boot: Up and Running*, O'Reilly Media, Second Edition, 2024.
5. Josh Long and Mark Heckler, *Spring Boot 3 Recipes: A Problem–Solution Approach*, Apress, Second Edition, 2024.
6. John Carnell and Illary Huaylupo Sánchez, *Spring Microservices in Action*, Manning Publications, Second Edition, 2021.
7. Rod Johnson, Juergen Hoeller, Keith Donald, Colin Sampaleanu, and Rob Harrop, *Professional Java Development with the Spring Framework*, Wiley Publishing, 2005. *(Recommended as a reference for understanding Spring Framework concepts.)*

Online Resources

1. Spring Framework Official Documentation
2. Spring Boot Official Documentation
3. Spring Guides
4. Spring Academy

5. Spring Initializr
6. Docker Documentation
7. JUnit 5 User Guide
8. Mockito Documentation
9. Postman Learning Center
10. Oracle Java Documentation

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510245041	Name of Course Cloud Computing	Compulsory
Credit: 02	Teaching Scheme: Theory (30) - Practical (0)	Teaching Hours: 30

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand cloud computing models and AWS architecture.

CO-2: Use core AWS services (compute, storage, networking, database).

CO-3: Apply security, monitoring, and cost optimization strategies.

CO-4: Develop or manage cloud-based solutions.

Detailed Syllabus

Unit-1. Cloud Fundamentals

1.2 Introduction to Cloud Computing
1.3 Evolution and need of cloud computing
1.4 Traditional vs cloud computing
1.5 Cloud Service Models (IaaS, PaaS, SaaS)
1.6 Cloud Deployment Models (Public, Private, Hybrid)
1.7 AWS Global Infrastructure
1.8 AWS Shared Responsibility Model
1.9 AWS Management Console overview

Unit-2. Core AWS Services

2.1 **Compute (EC2):** Instance types and pricing, AMI concepts, Key pairs and security groups, Launching and connecting to instances
2.2 **Storage**: Amazon S3: Buckets, objects, storage classes, Access permissions and policies, Elastic Block Store (EBS): Volume types and snapshots
2.3 **Networking:** Virtual Private Cloud (VPC) basics, Subnets and routing, Internet Gateway
2.4 **Database:** Amazon RDS: Instance creation and connectivity, DynamoDB basics
2.5 **Security:** Identity and Access Management (IAM): Users, roles, policies
2.6 **Practical**: EC2 instance setup and configuration, S3 bucket creation and static website hosting, IAM user and role configuration, Basic VPC setup, File storage and access management

Unit-3. Security, Billing & Monitoring

3.2 IAM deep dive
3.3 Multi-Factor Authentication (MFA)
3.4 AWS security best practices
3.5 AWS pricing models
3.6 Cost estimation and optimization
3.7 Budget setup and alerts
3.8 Amazon CloudWatch basics
3.9 Logs and alarms

Unit-4. AWS for Application Development

4.1 Application hosting using EC2
4.2 Elastic Beanstalk overview
4.3 AWS Lambda (serverless computing)
4.4 API Gateway basics
4.5 Database integration (RDS, DynamoDB)

4.6 Amazon Cognito (authentication basics)

4.7 CI/CD concepts and CodePipeline overview

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	3	-	2	-	-	-	2	-	2	-	-	1	2	2	-	--	1
CO-2	3	3	3	-	2	-	-	3	2	2	-	-	1.5	3	3	2	-	2
CO-3	2	3	-	-	3	3	-	2	-	2	-	-	1.25	2	3	2	2	2.25
CO-4	2	2	3	2	2	2	3	3	3	2	3	2	2.41	3	3	3	3	3
Avg	2.5	2.75	1.5	1	1.75	1.25	0.75	2.5	1.25	2	0.75	0.5	1.54	2.5	2.75	1.75	2.5	2.06

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Chalk & Talk (fundamental concepts), PPT-based explanation (service & deployment models), Real-life analogies (electricity billing → cloud usage), Short videos (intro to cloud)
CO-2 (Unit: 2)	• Demonstration-based teaching (EC2, S3, IAM), Guided walkthroughs, Case-based examples (host a website)
CO-3 (Unit: 3)	• Scenario-based teaching , Case discussions, Problem-solving sessions
CO-4 (Unit: 4)	• Case study method, Mini architecture discussions, Group brainstorming

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Rajkumar Buyya – Cloud Computing: Principles and Paradigms
- Thomas Erl – Cloud Computing Concepts
- AWS Training Documentation (Official)
- Distributed Systems & Cloud Architecture references.

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510345042	Name of Course Blockchain Technology	Elective-I
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the concepts of digital currencies, e-wallets, and cryptographic techniques such as hash functions and digital signatures, and differentiate between centralized, decentralized, and distributed systems in the context of blockchain.

CO-2: Evaluate how blockchain technology functions as a distributed ledger system, analyze different consensus algorithms, and implement basic smart contracts and decentralized applications (DApps) using Ethereum and Solidity.

CO-3: Describe the structure and functioning of the Bitcoin blockchain, assess Bitcoin's approach to decentralization, and analyze mining mechanisms, including their economic and ecological impacts.

CO-4: Explore the applications of blockchain technology in financial services, supply chain, and decentralized platforms, and demonstrate understanding of advanced frameworks such as Hyperledger Fabric, Corda, and IPFS.

Detailed Syllabus

Unit-1.

1.1. **Digital currency:** Digital currency and its Introduction. Crypto currency. Virtual currency. E-wallets – types, examples and working. Cryptography: Hash function, Digital Signature – ECDSA, Memory Hard Algorithm, Zero Knowledge Proof. Permission-less Blockchain

1.2. **Introduction to Blockchain:** Introduction to Blockchain Technology. Idea of Centralized, Decentralized and Distributed system, Blockchain as a Public ledger. Problems with a centralized system.

Unit-2.

2.1. **Introduction to Blockchain:** How Blockchain as a distributed ledger solve this problem. Advantage over conventional distributed database. Consensus models – concept. Consensus Algorithms – PoW, PoS, PBFT, DpoS, PoA, PoET. Comparative study of Consensus Algorithms

2.2. **Ether:** Ethereum, EthereumBlockchain and smart contracts, solidity, Dapps. Permissioned Blockchain

Unit-3.

3.1 **BitCoin:** Introduction to Bitcoin. Working of Bitcoin Blockchain. How Bitcoin achieve Decentralization (Distributed consensus) Bitcoin transactions, Bitcoin blocks, Bitcoin scripts, Bitcoin Network, Limitation & improvements. How to store and use Bitcoins – Hot and cold storage, online wallets and Exchanges, payments services, transaction fees, currency exchange market.

3.2 **Mining Techniques:** Bitcoin Mining the tasks of bitcoin miners, Mining hardware, Energy consumption & Ecology, Mining pools, Mining incentives and strategies. Types of Blockchain and its use cases and limitations.

Unit-4.

4.1 **Blockchain in Financial services:** Payments and Securities Trading – cross border payments, Stellar protocol and network, Ripple protocol and network. Logistics. Supply chain. Introduction, Hyperledger, Fabric services, Fabric model & functions, Composer, Corda. Decentralized Application Platforms. Alternative Decentralized Solutions – Interplanetary File Systems (IPFS), Hashgraph

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	2	1	2	1	2	1	2	–	2	1.82	3	2	2	2	2.25
CO-2	3	3	3	2	1	2	1	3	2	2	2	2	2.17	3	3	2	2	2.5
CO-3	3	2	2	2	2	2	–	3	1	2	–	2	2.10	3	2	2	2	2.25
CO-4	3	3	2	3	2	2	1	3	2	2	2	3	2.33	3	3	3	2	2.75
Avg	3	2.50	2.25	2.25	1.50	2	1	2.75	1.50	2	2	2.25	2.11	3	2.5	2.25	2	2.44

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	● Problem Solving, Examples, Questioning, Experiment, Demonstration, Seminar

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Bitcoin and Cryptocurrency Technologies by Arvind Narayanan, Joseph Bonneau, EdwardFelten, Andrew Miller and Steven Goldfeder, Princeton University Press, 2016.
- Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly Media, Inc., 2014.
- The Science of the Blockchain by Wattenhofer, Inverted Forest Publishing, 2016.
- Blockchain: The Blockchain for Beginners Guide to Blockchain Technology and Leveraging Blockchain Programming by Josh Thompson, CreateSpace Publishing, 2017

Online Resources & Tools:

- Mastering Ethereum by Antonopoulos, Andreas M. and Gavin Wood, O'Reilly Media, Inc., 2018. (Free draft available at https://github.com/ethereumbook/ethereumbook)
- **Hyperledger Fabric, https://www.hyperledger.org/use/fabric**

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510345043	Name of Course Agentic AI & Autonomous Intelligent Systems	Elective-II
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the evolution of Artificial Intelligence, intelligent agent architectures, Large Language Models (LLMs), prompt engineering techniques, and responsible AI principles for developing autonomous intelligent systems.

CO-2: Design intelligent agents using modern agent architectures, planning strategies, memory systems, Retrieval-Augmented Generation (RAG), and external tool integration techniques.

CO-3: Develop collaborative multi-agent systems and orchestrated workflows using modern agent development frameworks, multimodal AI concepts, and enterprise AI practices.

CO-4: Design, evaluate, and deploy trustworthy intelligent agent solutions by integrating reasoning, memory, workflows, evaluation metrics, and responsible AI governance principles.

Detailed Syllabus

Unit-1. Foundations of AI & Intelligent Agents

1.1. **Evolution of Artificial Intelligence:** History of AI • Symbolic AI • Expert Systems • Machine Learning • Deep Learning • Generative AI • Emergence of Agentic AI • Major milestones including Turing Test, IBM Deep Blue, AlphaGo, GPT, ChatGPT and modern autonomous AI systems • Narrow AI, General AI and Agentic AI • Current trends and future directions of intelligent systems.

1.2. **Intelligent Agents:** Concepts and Foundations Definition of intelligent agents • Characteristics of autonomous systems • Rational agents • Agent lifecycle • PEAS framework (Performance, Environment, Actuators and Sensors) • Types of intelligent agents: Simple Reflex, Model-Based, Goal-Based, Utility-Based, Learning Agents and BDI Agents • Agent environments • Problem solving through search • Introduction to heuristic search and reinforcement learning concepts.

1.3. **Large Language Models (LLMs):** Introduction to foundation models • Transformer architecture intuition • Tokenization • Self-attention mechanism • Context windows • Pre-training • Instruction tuning • Fine-tuning concepts • RLHF overview • Capabilities and limitations of modern LLMs • Comparative overview of GPT, Claude, Gemini, Llama, Mistral and other open-source models.

1.4. **Anatomy of an AI Agent:** AI Model versus AI System versus AI Agent • Agent architecture • Perception • Reasoning • Planning • Memory • Tool usage • Decision making • Goal-oriented behaviour • Agent execution cycle • Chatbots versus AI Assistants versus Autonomous Agents • Single-agent workflows.

1.5. **Reasoning and Prompt Engineering Fundamentals:** Prompt engineering principles • Zero-shot • One-shot • Few-shot prompting • Chain-of-Thought • Tree-of-Thought • Self-consistency • Reflection • Self-correction • Hallucination concepts • Prompt quality evaluation • Prompt engineering best practices.

1.6. **Responsible AI Fundamentals:** Bias and fairness • Transparency • Explainability • Privacy • Ethical AI principles • Human oversight • AI risk awareness • Introduction to AI governance • Responsible deployment of intelligent systems.

Unit-2. LLMs, Agent Architectures & Memory Systems

2.1. Core Agent Architectures: Introduction to modern agent architectures • Reactive versus Deliberative agents • ReAct (Reason + Act) architecture • Plan-and-Execute architecture • BDI (Belief–Desire–Intention) model • MRKL (Modular Reasoning, Knowledge and Language) systems • Reflexion architecture and self-improving agents • Agent execution loop • Goal management • Agent lifecycle • Selecting suitable architecture based on application requirements.

2.2. **Planning, Task Decomposition & Goal Management:** Problem decomposition • Hierarchical task decomposition • Sequential and parallel task execution • Goal prioritization • Sub-goal generation • Dynamic replanning • Handling uncertainty and incomplete information • Managing long-running tasks • Time-aware planning concepts • Human-in-the-loop intervention during planning.

2.3. **Memory Architecture & Knowledge Retrieval:** Need for memory in AI agents • Working memory, episodic memory, semantic memory and long-term memory • Context management • Embeddings and semantic similarity • Vector representations of text • Introduction to vector databases • ChromaDB, Pinecone, Weaviate, Qdrant and pgvector (overview) • Retrieval-Augmented Generation (RAG) architecture • Document chunking • Embedding generation • Similarity search • Hybrid retrieval concepts • Knowledge-grounded reasoning • Memory management strategies.

2.4. **Tool Use, Function Calling & External Integration:** Why AI agents require external tools • Tool selection strategies • Function calling concepts • Structured outputs using JSON schema • API fundamentals • REST APIs • Working with JSON • Authentication using API Keys • Browser tools • Calculator tools • File system tools • Code execution tools • Error handling and retry mechanisms • Tool orchestration concepts.

2.5. Prompt Engineering for Agentic Systems: System prompts • Persona engineering • Context engineering • Instruction hierarchy • Output formatting • Structured prompting • Meta prompting • Prompt templates • Prompt evaluation • Prompt optimization • Introduction to Model Context Protocol (MCP) • Prompt security basics including prompt injection awareness and defensive prompting techniques.

2.6. **Designing Knowledge-Based Intelligent Agents:** Integrating reasoning, memory and external tools into a complete intelligent agent • Designing agent workflows • Choosing memory strategies • Selecting tools for different applications • Balancing reasoning and retrieval • Performance considerations • Design patterns for research assistants, coding assistants, educational tutors and enterprise assistants.

Unit-3. Agent Frameworks, Multi-Agent Systems & Workflow Orchestration

3.1. Multi-Agent System Design: Introduction to Multi-Agent Systems (MAS) • Advantages of multi-agent collaboration • Distributed intelligence • Agent specialization • Orchestrator–Worker architecture • Planner–Executor pattern • Critic and Verifier agents • Reviewer agents • Peer-to-peer collaboration • Cooperative versus competitive agents • Agent communication models • Shared memory and private memory • Message passing concepts • Coordination strategies • Consensus mechanisms • Designing scalable collaborative agent systems.

3.2. Agent Development Frameworks & Platforms: Overview of modern agent development ecosystems • LangChain architecture and components • LangGraph for stateful agent workflows • LlamaIndex for knowledge-centric AI applications • CrewAI for role-based collaborative agents • AutoGen for conversational multi-agent systems • Microsoft Semantic Kernel overview • Comparative study of popular frameworks • Selection criteria based on application requirements • Framework interoperability • Future trends in agent development platforms.

3.3. Workflow Orchestration & State Management: Need for workflow orchestration • Sequential and parallel workflows • Directed Acyclic Graph (DAG) concepts • Stateful workflows • Agent state management • Conditional execution • Iterative execution loops • Task delegation • Human-in-the-Loop (HITL) workflows • Checkpointing and workflow recovery • Event-driven execution • Monitoring workflow progress • Designing reliable agent pipelines.

3.4. **Multimodal & Domain-Specific Intelligent Agents:** Introduction to multimodal AI • Text, image, audio and document understanding • Vision-language models • Document processing agents • Coding assistants • Research assistants • Educational assistants • Customer support agents • Healthcare assistants • Financial advisory assistants • Legal document assistants • Enterprise knowledge assistants • Integrating AI agents with databases, ERP and CRM systems (conceptual overview).

3.5. **Enterprise Essentials for Intelligent Agents:** Introduction to AI deployment lifecycle • Concept of LLMOps and AgentOps • Prompt security • Prompt injection attacks • Jailbreak awareness • Guardrails for AI systems • Secure tool usage • Data privacy fundamentals • Responsible handling of sensitive information • Audit trails • Explainability • Introduction to ISO/IEC 42001, NIST AI Risk Management Framework and Responsible AI governance concepts.

3.6. **Design Patterns for Intelligent Agent Applications:** Designing AI agents for research automation • Knowledge management • Education • Software development • Business process automation • Customer support • Report generation • Decision support • Selecting suitable frameworks, memory models, tools and orchestration strategies • Case studies illustrating end-to-end intelligent agent solutions.

Unit-4. Integrated Agent Design, Evaluation & Responsible AI

4.1. **End-to-End Intelligent Agent Design Methodology:** Identifying business problems suitable for AI agents • Requirement analysis • Defining agent objectives and success criteria • PEAS-based problem modelling • Selecting appropriate agent architecture • Choosing reasoning strategies • Selecting memory architecture • Tool selection and integration • Designing prompt architecture • Workflow modelling • Performance considerations • Documentation of agent design decisions • Best practices for modular and maintainable intelligent agent development.

4.2. **Building Integrated Intelligent Agent Solutions:** Designing complete AI agent workflows by integrating reasoning, planning, memory, Retrieval-Augmented Generation (RAG), external tools and APIs • Single-agent versus multi-agent implementation strategies • Human-in-the-Loop (HITL) integration • Exception handling • Fallback mechanisms • Conversation management • Agent monitoring concepts • Scalability considerations • Case studies of end-to-end intelligent agent applications.

4.3. **Evaluation, Testing & Quality Assurance of AI Agents:** Need for systematic evaluation • Functional and non-functional evaluation • Task completion metrics • Accuracy, relevance and faithfulness • Hallucination detection • Robustness and reliability • Human evaluation methodologies • LLM-as-Judge concepts • Introduction to GAIA, AgentBench and other benchmark datasets • Test case design • Performance reporting • Continuous improvement of intelligent agents.

4.4. **Responsible AI, Safety & Governance:** AI ethics in intelligent systems • Fairness and bias mitigation • Explainability and interpretability • Privacy-preserving AI • Secure use of external tools • Transparency and accountability • Human oversight • AI risk assessment • Responsible AI lifecycle • Overview of AI governance frameworks including ISO/IEC 42001, NIST AI RMF and emerging regulatory principles • Professional responsibilities in AI development.

4.5. **Applications, Case Studies & Future Directions:** Applications of Agentic AI in education, healthcare, finance, legal services, software engineering, manufacturing, logistics, research, e-commerce and public services • Emerging trends in autonomous AI systems • Intelligent digital assistants • Collaborative AI ecosystems • Future of human–AI collaboration • Career opportunities in Agentic AI and Intelligent Systems • Challenges and research directions.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	2	–	2	–	3	–	3	–	1	1.5	3	3	3	3	3
CO-2	3	3	3	2	–	1	–	3	1	2	1	1	1.67	3	3	2	1	2.25
CO-3	3	3	3	2	–	1	1	3	3	3	2	1	2.1	3	3	2	2	2.5
CO-4	2	2	2	1	1	3	1	2	1	2	1	2	1.67	2	2	2	3	2.25
Avg	2.75	2.5	2.5	1.75	0.25	1.75	0.5	2.75	1.25	2.5	1	1.25	1.72	2.75	2.75	2.25	2.25	2.5

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Problem Solving, Examples, Questioning, Assignment, Experimental, Demonstration, Project, Activity

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Primary Texts:

- Stuart Russell & Peter Norvig — Artificial Intelligence: A Modern Approach (4th Edition)
- Chip Huyen — AI Engineering
- Christopher Manning, Jacob Eisenstein & others — Natural Language Processing with Deep Learning
- Michael Wooldridge — An Introduction to MultiAgent Systems
- Jurafsky & Martin — Speech and Language Processing (Latest Edition)

Research Papers & Technical Reading

- ReAct: Synergizing Reasoning and Acting in Language Models
- Reflexion: Language Agents with Verbal Reinforcement Learning
- Retrieval-Augmented Generation (RAG)
- LangGraph Architecture
- Tree-of-Thought Prompting
- Chain-of-Thought Prompting
- Self-Consistency Reasoning
- AgentBench Benchmark
- GAIA Benchmark

Framework Documentation

- LangChain
- LangGraph
- CrewAI
- AutoGen
- LlamaIndex
- Microsoft Semantic Kernel
- OpenAI API
- Anthropic Claude API
- Google Gemini API

Suggested Online Learning Resources

- Hugging Face Documentation
- DeepLearning.AI Short Courses
- OpenAI Learning Resources
- Anthropic Learning Center
- Microsoft Learn – AI
- Google AI Learning Resources

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510345044	Name of Course Advanced Database Management System	Elective-I
Credit: 03	Teaching Scheme: Theory (45) - Practical (0)	Teaching Hours: 45

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand and define a database instance, describe its components, and design a flexible and optimal database architecture for various administrative requirements.

CO-2: Perform data export/import operations using Oracle utilities, load data from third-party systems, and write SQL stored programs including procedures and packages.

CO-3: Manage database storage, implement and monitor locking mechanisms, and use supplied Oracle packages to address storage and concurrency issues.

CO-4: Apply performance tuning techniques for SQL statements, memory components, and I/O operations using appropriate tools and strategies.

Detailed Syllabus

Unit-1.

1.1 **The Database Instance and Database Architecture:** Defining the Instance, Creating the Instance, Understanding the Instance, Defining the Database, Different schema for system administration, Understanding the Components of the Database, Understanding Database Segments, Other Database Objects, Creating the Environment, Designing an Optimal Flexible Architecture, Creating Database

Unit-2.

2.1 **Export-Import & loading data from third party s/w, SQL with Programing**

2.2 **IMPORT/EXPORT:** Export with Data pump utility, Import with Data pump utility, Loading data from third party database With various cases like truncation of data , conversion of data, filtering data with where clause, Different Loader Examples, Conventional and Direct Path Loading

2.3 **Administering SQL commands:** Create, Alter, Truncate, Drop etc. DDL and DCL SQL commands, Tracing SQL Statements

2.4 **SQL Programming:** Stored subprograms and packages, Defining Stored Subprograms., Building and Using Stored Programs

Unit-3.

3.1 **Managing Database Storage:** Administering Database Objects, Understanding Database Fragmentation, Managing Rollback Segments, Identifying Storage Problems, Administering Growing Database, Integrity Management

3.2 **Locking:** Implementing Locks, Analyzing lock table, Monitoring Locks on the System, Avoiding Locks & Possible Solutions, Implementing Locks with Latches

Unit-4.

4.1 **Performance tuning fundamentals:** Understanding need of tuning, Knowing the Tuning Principles, Tuning Goals, Using the Return on Investment Strategy, Revisiting Application Types, Using Diagnostic Tools.

4.2 **Application Tuning:** Understanding the Optimizer, SQL Trace and derivation of statistics, Understanding execution plan

4.3 **Tuning Memory:** UTLBSTAT/UTLESTAT, Tuning the Shared Pool, Tuning the Database Buffer Cache - Tuning the multithreaded Server (MTS), Tuning Locks

4.4 **Tuning I/O:** Tuning Table-spaces and Data-files, Tuning Blocks and Extents, Tuning Rollback Segments, Tuning Redo Logs

Mapping Matrix of POs, PSOs, and Cos

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	1	–	–	1	2	1	1	–	–	1.58	3	2	1	1	1.75
CO-2	3	2	3	–	–	–	1	2	1	1	1	–	1.67	3	2	1	2	2
CO-3	3	3.	3	–	–	–	–	2	1	1	1	–	1.67	3	2	2	2	2.25
CO-4	2	3	3	–	–	–	–	3	1	2	2	–	2	3	3	2	2	2.5
Avg	2.75	2.5	3	0.25	–	–	0.75	2.25	1	1.25	1	–	1.73	3	2.25	1.5	1.75	2.13

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Classroom Teaching, Group Discussion, Assignment, Examples, Questioning

Assessment Method

	COs	Marks	Exam Component			
			Class Test	Assignment	Presentation	Involvement
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	5	7	7	6
	CO-2	10	5			
	CO-3	10	5			
	CO-4	10	5			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component			
	CO-1	15	Term End Examination			
	CO-2	15				
	CO-3	15				
	CO-4	15				

References

Books:

- Rajeev Parida, Oracle 10g Performance Tuning, Firewall media
- Rajeev Parida, The Power of Oracle 10g, Firewall Media
- Craig S. Mullin, Database Administration: The Complete Guide to DBA Practices and Procedures, Oracle Press
- Oracle Complete Reference, Oracle Press
- Oracle DBA, Oracle Press

Online Resources & Tools:

- Oracle Manual from www.oracle.org

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510245046	Name of Course Lab Based on Microservices using Spring Boot	Compulsory
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the principles of software architecture, microservices, Spring Framework, Spring Boot, and RESTful web services for developing modern enterprise applications.

CO-2: Develop RESTful web applications using Spring Boot by applying Spring Core, dependency injection, Spring MVC, and application configuration techniques.

CO-3: Design and implement database-driven applications using Spring Data JPA by performing entity mapping, repository development, CRUD operations, and transaction management.

CO-4: Develop secure, tested, and deployable Spring Boot applications by applying Spring Security, software testing techniques, and container-based deployment concepts.

Mapping Matrix of POs, PSOs, and Cos

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	-	-	1	2	-	1	-	-	1.00	3	2	-	-	1.25
CO-2	3	2	3	-	-	1	-	3	2	1	-	-	1.25	3	3	-	-	1.50
CO-3	3	2	3	-	-	1	-	3	1	1	-	-	1.17	3	3	-	-	1.50
CO-4	3	2	3	-	1	3	1	3	2	2	-	1	1.75	3	3	1	2	2.25
Avg	3	2	2.75	0.25	0.25	1.25	0.50	2.75	1.25	1.25	0	0.25		3	2.75	0.25	1.00	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Problem-Based Learning , Case-Based Learning , Individual Practice Exercises • Live Coding Sessions , Problem-Solving Exercises • Database-Driven Application Development , Guided Laboratory Exercises • Testing and Debugging Workshops , Mini Project Development , Deployment Demonstration

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal/Project Demonstration	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

References

Books

1. Craig Walls, *Spring Boot in Action*, Manning Publications, First Edition, 2016. *(Foundational reference; although older, it covers Spring Boot fundamentals well.)*
2. Laurentiu Spilca, *Spring Start Here: Learn What You Need and Learn It Well*, Manning Publications, Second Edition, 2024.
3. Laurentiu Spilca, *Spring Security in Action*, Manning Publications, Second Edition, 2024.
4. Mark Heckler, *Spring Boot: Up and Running*, O'Reilly Media, Second Edition, 2024.
5. Josh Long and Mark Heckler, *Spring Boot 3 Recipes: A Problem–Solution Approach*, Apress, Second Edition, 2024.
6. John Carnell and Illary Huaylupo Sánchez, *Spring Microservices in Action*, Manning Publications, Second Edition, 2021.
7. Rod Johnson, Juergen Hoeller, Keith Donald, Colin Sampaleanu, and Rob Harrop, *Professional Java Development with the Spring Framework*, Wiley Publishing, 2005. *(Recommended as a reference for understanding Spring Framework concepts.)*

Online Resources

1. Spring Framework Official Documentation
2. Spring Boot Official Documentation
3. Spring Guides
4. Spring Academy
5. Spring Initializr
6. Docker Documentation
7. JUnit 5 User Guide
8. Mockito Documentation
9. Postman Learning Center
10. Oracle Java Documentation

Program - MCA (Faculty of Information, Communication and Technology)
Semester- 3

Course Code 266510145047	Name of Course Lab based on Cloud Computing	
Credit: 01	**Teaching Scheme: Theory (0) - Practical (30)**	**Lab Hours: 30**

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Perform basic operations on AWS platform (EC2, S3, IAM)

CO-2: Deploy and manage cloud-based resources

CO-3: Implement security and monitoring configurations

CO-4: Develop and deploy a cloud-based application

Detailed Syllabus

Unit-1. Cloud Fundamentals

1.1. Introduction to AWS environment / AWS Educate
1.2. AWS Console navigation and service overview

Unit-2. Core AWS Services

2.1. EC2 instance setup and configuration.
2.2. S3 bucket creation and static website hosting.
2.3. Removing duplicates, handling missing data, and transforming raw data into a structured format.
2.4. Generating summary statistics and initial reports from cleaned data.

Unit-3. Security, Billing & Monitoring

3.1. IAM policy configuration
3.2. Budget and billing monitoring (demo/simulation).
3.3. CloudWatch monitoring and alerts

Unit-4. AWS for Application Development

4.1. Deploy a web application on EC2.
4.2. Database integration with the application.
4.3. Basic serverless implementation (Lambda – conceptual/lab)
4.4. Mini Project: Cloud-based application deployment

Mapping Matrix of POs, PSOs, and Cos

	POs													PSOs				
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	11	**12**	**Avg**	**1**	**2**	**3**	**4**	**Avg**
CO-1	2	2	3	-	-	-	-	3	-	2	-	-	**1**	3	3	-	-	**1.5**
CO-2	2	3	3	-	2	-	-	3	2	2	-	-	**1.5**	3	3	2	-	**2**
CO-3	3	3	2	-	3	3	-	2	-	2	3	2	**1.58**	2	3	2	2	**2.25**
CO-4	2	2	3	2	2	2	3	3	3	2	-	2	**2.41**	3	3	3	3	**3.00**
Avg	**2.25**	**2.5**	**2.75**	**0.5**	**1.75**	**1.25**	**0.75**	**2.75**	**1.25**	**2**	0.75	**1**	**1.62**	**3.0**	**3.0**	**1.75**	**1.25**	**2.18**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Demonstration + hands-on practice, Step-by-step lab guidance
CO-2 (Unit: 2)	• Task-based learning, Guided lab exercises
CO-3 (Unit: 3)	• Scenario-based lab tasks, Troubleshooting exercises
CO-4 (Unit: 4)	• Mini project-based learning, Peer collaboration, Faculty mentoring

<table>
<tr><th colspan="6">Assessment Method</th></tr>
<tr><td rowspan="6">Continuous Comprehensive Evaluation 40 Marks</td><td rowspan="2">COs</td><td rowspan="2">Marks</td><td colspan="3">Exam Component</td></tr>
<tr><td>Practical Test</td><td>Journal</td><td>Viva</td></tr>
<tr><td>CO-1</td><td>10</td><td rowspan="2">10</td><td rowspan="3">13</td><td rowspan="3">7</td></tr>
<tr><td>CO-2</td><td>10</td></tr>
<tr><td>CO-3</td><td>10</td><td rowspan="2">10</td></tr>
<tr><td>CO-4</td><td>10</td></tr>
<tr><td rowspan="5">Term-End Evaluation 60 Marks</td><td>COs</td><td>Marks</td><td colspan="3">Exam Component</td></tr>
<tr><td>CO-1</td><td>15</td><td colspan="3" rowspan="4">Term End Examination</td></tr>
<tr><td>CO-2</td><td>15</td></tr>
<tr><td>CO-3</td><td>15</td></tr>
<tr><td>CO-4</td><td>15</td></tr>
</table>

References

Books:

- Cloud Computing: A Practical Approach - Toby Velte, Anthony Velte, Robert Elsenpeter
- Cloud Computing Basics - Anders Lisdorf
- AWS Certified Cloud Practitioner Study Guide - Ben Piper

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510245048	Name of Course Lab based on Blockchain Technology	Elective-I
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand the concepts of digital currencies, cryptographic tools, and decentralized systems fundamental to blockchain.

CO-2: Analyze blockchain architecture, consensus mechanisms, and develop smart contracts on permissioned and permission-less platforms.

CO-3: Examine the Bitcoin ecosystem including transaction models, storage, mining techniques, and evaluate its technical limitations.

CO-4: Apply blockchain technologies in real-world applications such as finance, logistics, and decentralized systems using Hyperledger, IPFS, and Corda.

Mapping Matrix of POs, PSOs, and Cos

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	2	1	2	–	3	1	2	–	2	2.10	3	2	2	2	2.25
CO-2	3	3	3	2	–	2	–	3	2	2	2	2	2.40	3	3	2	2	2.5
CO-3	3	2	3	–	–	2	–	3	1	2	1	1	2.00	3	2	2	2	2.25
CO-4	3	2	3	2	2	2	1	3	2	2	2	3	2.25	3	3	2	3	2.75
Avg	3	2.25	3	2	1.50	2	1	3	1.50	2	1.67	2	2.19	3	2.5	2	2.25	2.44

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Experimental Learning
CO-2 (Unit: 2)	• Demonstration and Guided Practice
CO-3 (Unit: 3)	• Project-Based Learning
CO-4 (Unit: 4)	• Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	COs	Marks	Exam Component		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)
Semester- 3

Course Code 266510245049	Name of Course Lab based on Agentic AI & Autonomous Intelligent Systems	Elective-II
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Pre-process and prepare data for machine learning by applying data cleaning, balancing, and transformation techniques.

CO-2: Implement and evaluate supervised learning models (K-NN, SVM, linear and logistic regression) using appropriate metrics.

CO-3: Apply unsupervised learning techniques (K-means, PCA) and analyze clustering/dimensionality reduction results.

CO-4: Demonstrate foundational understanding of AI through implementation of basic ANN models including perceptron and MLP.

Detailed Syllabus

Unit-1. Foundations Lab

1.1 Python AI Environment Setup & LLM API Basics
1.2 Transformer Internals — Tokenisation & Attention Visualisation
1.3 Mapping Agent Anatomy — PEAS & Agent Type Classifier
1.4 Prompt Engineering Laboratory — Reasoning Pattern Experiments
1.5 AI Ethics Audit — Bias Detection & Hallucination Analysis

Unit-2. Agent Architecture Lab

2.1. Implementing the ReAct Agent Loop from Scratch
2.2. Plan-and-Execute Agent with Task Decomposition
2.3. Building a RAG Pipeline — Naive to Advanced
2.4. Vector Database Comparative Study — ChromaDB vs Qdrant vs pgvector
2.5. Structured Tool Use & Function Calling — Multi-Tool Agent
2.6. Prompt Engineering for Agents — System Prompt Architecture Lab

Unit-3. Multi-Agent & Frameworks Lab

3.1. Multi-Agent Design Patterns — Orchestrator & Critic Roles
3.2. LangGraph — Stateful Agent Graphs with Cycles & Conditionals
3.3. CrewAI — Role-Based Agent Crew for Content Production
3.4. AutoGen — Conversational Multi-Agent Debate & Consensus
3.5. Workflow Orchestration — DAG Pipeline with Error Recovery
3.6. Multimodal Agent — Vision + Language + Code Execution
3.7. Agent Evaluation Framework — LLM-as-Judge & Benchmark Harness

Unit-4. Enterprise AI Lab

4.1. LLMOps Pipeline — Serving, Monitoring & Cost Optimisation
4.2. LoRA Fine-tuning — Domain Adaptation on Small Dataset
4.3. Dataset Curation & Preference Data for Alignment
4.4. AI Safety Lab — Prompt Injection, Jailbreaks & Guardrails
4.5. Observability Stack — LangSmith Tracing & Agent Monitoring Dashboard

Mapping Matrix of POs, PSOs, and Cos

COs	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	2	2	3	1	–	2	1	3	1	2	–	1	1.5	2	3	1	2	2
CO-2	3	3	3	2	–	1	1	3	2	2	1	1	1.8	3	3	2	1	2.25
CO-3	3	3	3	2	–	1	1	3	3	3	1	1	2	3	3	3	2	2.75
CO-4	3	2	3	1	1	3	1	3	2	3	1	2	2.1	3	3	2	2	2.5
Avg	2.75	2.5	3	1.5	0.25	1.75	1	3	2	2.5	0.75	1.25	1.85	2.75	3	2	1.75	2.38

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1)	• Experimental Learning, Demonstration and Guided Practice, Project-Based Learning, Reflective Practice
CO-2 (Unit: 2)	
CO-3 (Unit: 3)	
CO-4 (Unit: 4)	

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
	COs	Marks	Exam Component		
Term-End Evaluation 60 Marks	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)
Semester- 3

Course Code 266510145050	Name of Course Lab based on Advanced Database Management System	Elective-I
Credit: 01	Teaching Scheme: Theory (0) - Practical (30)	Lab Hours: 30

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Understand the fundamental components of a database system and implement a well-structured database instance, apply DDL, DCL and DML.

CO-2: Perform advanced import/export operations, load data efficiently from external sources, and develop stored procedures and packages.

CO-3: Manage and optimize database storage and locking mechanisms using built-in administrative tools and packages.

CO-4: Apply core performance tuning techniques to enhance database efficiency through memory, I/O, and SQL optimization strategies.

Mapping Matrix of POs, PSOs, and COs

	POs													PSOs				
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Avg**	**1**	**2**	**3**	**4**	**Avg**
CO-1	3	–	2	–	–	1	1	2	–	1	–	–	**1.18**	3	2	1	1	**1.75**
CO-2	3	–	3	–	–	1	–	2	–	1	1	–	**1.55**	3	3	1	1	**2**
CO-3	3	–	3	–	1	2	–	3	–	1	–	–	**1.73**	3	3	2	1	**2.25**
CO-4	3	–	3	–	1	2	–	3	–	2	–	–	**1.91**	3	3	2	2	**2.5**
Avg	**3**	**–**	**2.75**	**–**	**0.5**	**1.5**	**0.25**	**2.5**	**–**	**1.25**	**0.25**	**–**	**1.59**	**3**	**2.75**	**1.5**	**1.25**	**2.13**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Experimental Learning, Demonstration and Guided Practice, Project-Based Learning, Reflective Practice

Assessment Method

	Cos	Marks	Exam Component		
			Practical Test	**Journal**	**Viva**
Continuous Comprehensive Evaluation 40 Marks	**CO-1**	10	10	13	7
	CO-2	10			
	CO-3	10	10		
	CO-4	10			
Term-End Evaluation 60 Marks	**Cos**	**Marks**	**Exam Component**		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510445052	Name of Course Internship /Minor Projects	Compulsory
Credit: 04	Teaching Scheme: Theory (0) - Practical (120)	Hours: 120

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Identify and analyse a real-life problem, gather user requirements, and define system specifications.

CO-2: Design and develop software applications using suitable tools, technologies, and development methodologies.

CO-3: Collaborate effectively within a team to manage the project lifecycle, including version control, testing, and deployment.

CO-4: Demonstrate professionalism through documentation, presentation, and awareness of the ethical, social, and environmental impacts of the software developed.

Mapping Matrix of POs, PSOs, and Cos

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	3	3	2	–	2	–	2	2	2	2	2	**2.27**	3	3	2	2	**2.5**
CO-2	3	2	3	2	2	–	–	3	–	–	2	–	**2.22**	3	3	2	2	**2.5**
CO-3	--	2	3	–	–	–	2	2	3	–	3	–	**2.14**	3	2	2	2	**2.25**
CO-4	2	–	2	2	2	3	3	–	–	2	–	3	**2.33**	2	2	3	3	**2.5**
Avg	**2.67**	**2.33**	**2.75**	**2**	**1.33**	**1.67**	**1.67**	**2.33**	**1.67**	**1.33**	**2.33**	**1.67**	2.24	**2.75**	**2.5**	**2.25**	**2.25**	**2.44**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Assessment Method

Evolution as per Minor project guideline

Appendix-A

Program – MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266520345053	Name of Course MOOC-Programming with Generative AI	MOOC-E3
Credit: 03 NCrF level: 4.5 – 8.0	Teaching Scheme: Theory (Online)	Teaching Hours: 8 Weeks

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Use Generative AI tools and techniques for assisting software development, enabling them to understand data representation, programming fundamentals, and AI-assisted code generation.

CO-2: Analyze and decompose complex computational problems into manageable tasks and utilize Generative AI effectively for designing, implementing, and debugging program solutions.

CO-3: Skills in writing structured, modular, and efficient programs through the use of functions, recursion, testing methodologies, and AI-supported code evaluation practices.

CO-4: Learn and adapt to new programming languages and technologies with the support of Generative AI, while understanding performance optimization, memory management, and best programming practices.

Detailed Syllabus

Week 1: Introduction: What is programming and how effective are modern GenAI tools at programming? Optional: Installing GenAI tools on your computer.

Week 2: Representing and manipulating data: Fundamental data types (and potential mismatches with a problem's needs), objects, expressions, assignments, and error handling in Python.

Week 3: Decomposing complex problems: Using functions to represent and solve sub-problems, principles of good function design. Introduction to conditionals and recursion in Python.

Week 4: Critiquing code: Comprehending and testing code (including AI-generated code) to identify potential errors. Introduction to lists and iteration in Python.

Week 5: Debugging problem statements and code: Asking clarifying questions, manual and AI-assisted debugging. Introduction to dictionaries in Python.

Week 6: Learning a new programming language (C): C vs Python: key differences. Introduction to arrays and structures in C.

Week 7: Introduction to memory management in C: A deeper understanding of Python lists by developing a simplified implementation of Python lists in C.

Week 8: Fast implementations: Identifying "hot" Python code and translating into C, a peek ahead into Data Structures and Algorithm.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	1	3	1	2	1	-	1	-	2	1	**2**	3	2	1	1	**2**
CO-2	2	3	3	3	2	3	1	1	1	-	2	2	**2**	3	3	1	2	**2**
CO-3	2	3	3	2	2	3	1	1	1	-	2	2	**2**	3	2	1	2	**2**
CO-4	2	2	2	3	2	2	1	1	1	1	3	2	**2**	2	2	2	2	**2**
Avg	**2**	**3**	**2**	**3**	**2**	**3**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**3**	**2**	**1**	**2**	**2**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

<table>
<tr><th colspan="4">Teaching Pedagogy</th></tr>
<tr><td>CO-1 (Unit: 1)
CO-2 (Unit: 2)
CO-3 (Unit: 3)
CO-4 (Unit: 4)</td><td colspan="3">● As per MOOC Instructor</td></tr>
<tr><th colspan="4">Assessment Method</th></tr>
<tr><td colspan="3" rowspan="2">Continuous Comprehensive Evaluation
40 Marks</td><td>Exam Component</td></tr>
<tr><td>AS per MOOC Guidelines</td></tr>
<tr><td rowspan="2">Term-End Evaluation
60 Marks</td><td>Case Study</td><td>Theory</td><td>Exam Component</td></tr>
<tr><td>20</td><td>40</td><td>Term End Examination</td></tr>
<tr><th colspan="4">References</th></tr>
<tr><td colspan="4">Books:
● Python 3 documentation (https://docs.python.org/3/)
● The GNU C Reference Manual (https://www.gnu.org/ software/gnu-c-manual/gnu-c-manual.html)
● Learn AI-Assisted Python Programming (Leo Porter and Daniel Zingaro, Manning Sept 2023, ISBN 9781633437784)
Online Resources & Tools:
● UGC Swayam Portal(Swayam Central)</td></tr>
<tr><td colspan="4">Course Link: https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs188
Course Instructor: Prof. Viraj Kumar, IISC Bangalore</td></tr>
</table>

Appendix-B

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266520345045	Name of Course MOOC-Big Data Computing	Elective-II
Credit: 03 NCrF level: 4.5 – 8.0	Teaching Scheme: Theory (Online)	Teaching Hours: 8 Weeks

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Explain the fundamental concepts, characteristics, enabling technologies, and architectures of Big Data systems and their significance in modern computing applications.

CO-2: Analyze and utilize Big Data platforms, storage frameworks, and streaming technologies for managing and processing large-scale and fast-moving data efficiently.

CO-3: Apply machine learning techniques and Spark-based frameworks to develop scalable data analytics solutions for real-world Big Data applications.

CO-4: Evaluate and implement Big Data applications, including graph processing and interdisciplinary use cases, to address complex industrial, research, and societal challenges.

Detailed Syllabus

Week 1 : Introduction to Big Data
Week 2 : Introduction to Enabling Technologies for Big Data
Week 3 : Introduction to Big Data Platforms
Week 4 : Introduction to Big Data Storage Platforms for Large Scale Data Storage
Week 5 : Introduction to Big Data Streaming Platforms for Fast Data
Week 6 : Introduction to Big Data Applications (Machine Learning)
Week 7 : Introduction of Big data Machine learning with Spark
Week 8 : Introduction to Big Data Applications (Graph Processing)

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	2	1	1	1	3	1	2	1	1	1.67	3	3	1	1	2.00
CO-2	3	3	3	2	2	1	1	3	2	2	2	2	2.17	3	3	2	1	2.25
CO-3	3	3	3	2	2	1	1	3	2	2	3	2	2.25	3	3	2	2	2.50
CO-4	3	3	3	3	2	2	2	3	2	2	3	3	2.58	3	3	3	2	2.75
Avg	3	2.75	2.75	2.25	1.75	1.25	1.25	3	1.75	2	2.25	2	2.17	3	3	2	1.50	2.38

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• As per MOOC Instructor

Assessment Method

	COs	Marks	Exam Component
Continuous Comprehensive Evaluation 40 Marks	CO-1		AS per MOOC Guidelines
	CO-2		
	CO-3		
	CO-4		

Term-End Evaluation 60 Marks	COs	Marks	Exam Component
	CO-1	15	Term End Examination
	CO-2	15	
	CO-3	15	
	CO-4	15	

References

Books:

- Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Wiley, 2014
- Dirk Deroos et al., Hadoop for Dummies, Dreamtech Press, 2014.
- Chuck Lam, Hadoop in Action, December, 2010.
- Leskovec, Rajaraman, Ullman, Mining of Massive Datasets, Cambridge University Press.
- I.H. Witten and E. Frank, Data Mining: Practical Machine learning tools and techniques.
- Erik Brynjolfsson et al., The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies, W. W. Norton & Company, 2014.

Online Resources & Tools:

- UGC Swayam Portal(Swayam Central)

Course Link: **https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs189**

Course Instructor: Prof. Rajiv Misra, IIT Patna

Appendix - C

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 3

Course Code 266510245054	Name of Course MOOC-Lab Based on Programming with Generative AI	MOOC
Credit: 02	Teaching Scheme: Theory (0) - Practical (60)	Lab Hours: 60

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Develop Python programs using fundamental programming constructs, including variables, data types, expressions, functions, conditionals, recursion, exception handling, and built-in data structures, while effectively utilizing Generative AI tools for code generation and learning.

CO-2: Analyze, test, debug, and improve manually written and AI-generated Python programs by applying systematic debugging techniques, code comprehension, iteration, lists, dictionaries, and software testing practices.

CO-3: Compare and implement core programming concepts in Python and C, including arrays, structures, memory management, and data representation, to understand language-specific design and performance considerations.

CO-4: Design and optimize computational solutions by identifying performance bottlenecks, translating critical Python code into C for improved efficiency, and applying foundational concepts of algorithmic thinking and data structures with appropriate use of Generative AI assistance.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	3	-	-	2	1	3	-	2	-	-	1.33	3	3	-	1	1.75
CO-2	2	3	3	-	-	3	2	2	1	3	-	-	1.58	3	2	2	3	2.5
CO-3	3	2	3	1	-	1	1	3	-	2	-	-	1.33	3	3	1	1	2
CO-4	3	3	3	2	1	2	1	3	2	3	2	1	1.92	3	3	2	2	2.5
Avg	2.75	2.5	3	1.5	1	2	1.25	2.75	1.5	2.5	2	1		3	2.75	1.67	1.75	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) CO-2 (Unit: 2) CO-3 (Unit: 3) CO-4 (Unit: 4)	• Experimental Learning, Demonstration and Guided Practice • Project-Based Learning, Reflective Practice

Assessment Method

	COs	Marks	Exam Component		
			Practical Test	Journal	Viva
Continuous Comprehensive Evaluation 40 Marks	CO-1		10	13	7
	CO-2				
	CO-3		10		
	CO-4				
Term-End Evaluation 60 Marks	COs	Marks	Exam Component		
	CO-1	15	Term End Examination		
	CO-2	15			
	CO-3	15			
	CO-4	15			

Appendix-D

Program – MCA (Faculty of Information, Communication and Technology)		
Semester- 3		
Course Code **266510145051**	**Name of Course** **MOOC-Case Study based on Big Data Computing**	**Elective-II**
Credit: 01	**Teaching Scheme: Theory (0) - Practical (30)**	**Teaching Hours: 30**

Course Outcomes (COs)

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

CO1: Explain the fundamental concepts, enabling technologies, and architecture of Big Data systems and distributed computing platforms.

CO2: Apply Big Data storage and processing frameworks, including Hadoop and Apache Spark, to manage and analyze large-scale datasets.

CO3: Design and implement real-time Big Data applications using streaming platforms and Spark-based data processing techniques.

CO4: Develop and evaluate scalable Big Data solutions using Spark MLlib and graph processing frameworks to solve real-world analytical problems.

Mapping Matrix of POs, PSOs, and COs

COs	POs													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	2	2	1	-	-	-	3	-	1	-	-	1.00	3	2	-	-	1..25
CO-2	3	2	3	1	-	-	-	3	1	1	-	-	1.17	3	3	-	-	1.50
CO-3	3	2	3	1	-	-	-	3	1	2	-	-	1.33	3	2	1	-	1.75
CO-4	3	3	3	2	-	1	-	3	-	2	-	-	1.50	3	3	2	-	2.00
Avg	3	2.25	3	1.50	0	0.25	0	3	0.75	1.50	0	0		3	2.75	0.75	0	

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Teaching Pedagogy

CO-1 (Unit: 1) **CO-2 (Unit: 2)** **CO-3 (Unit: 3)** **CO-4 (Unit: 4)**	• Experimental Learning, Demonstration and Guided Practice • Project-Based Learning, Reflective Practice

Assessment Method

Continuous Comprehensive Evaluation 40 Marks	**COs**	**Marks**	**Exam Component**
			AS per MOOC Guidelines
	CO-1		
	CO-2		
	CO-3		
	CO-4		
Term-End Evaluation 60 Marks	**COs**	**Marks**	**Exam Component**
	CO-1	15	Term End Examination
	CO-2	15	
	CO-3	15	
	CO-4	15	

Program - MCA (Faculty of Information, Communication and Technology)

Semester- 4

Course Code 266512045032	Name of Course Internship	Compulsory
Credit: 20	Teaching Scheme: Theory (0) - Practical (600)	Hours: 600

Course Outcomes (COs)

After completing this course, students will be able to

CO-1: Students will be able to analyze user requirements and design software solutions that address real-world problems, particularly those relevant to local, rural, or national contexts.

CO-2: Students will be able to apply advanced programming concepts, software engineering principles, and modern tools/technologies to implement, test, and deploy software systems.

CO-3: Students will be able to demonstrate project management, teamwork, and leadership skills while collaboratively developing and delivering software solutions in alignment with ethical, sustainable, and socially responsible practices.

CO-4: Students will be able to document and communicate project deliverables effectively through technical reports, presentations, and demonstrations in both English and Gujarati, showcasing the functionality and societal impact of their work.

Mapping Matrix of POs, PSOs, and COs

Cos	Pos													PSOs				
	1	2	3	4	5	6	7	8	9	10	11	12	Avg	1	2	3	4	Avg
CO-1	3	3	3	3	2	2	-	2	2	1	2	3	**2.27**	3	2	2	2	**2.25**
CO-2	3	2	3	1	2	2	-	3	1	2	1	1	**1.83**	3	2	2	1	**2**
CO-3	1	2	2	2	2	3	-	1	3	2	3	2	**2**	3	3	2	2	**2.5**
CO-4	--	-	-	-	-	1	3	-	-	-	-	2	**2**	2	2	1	1	**1.5**
Avg	**2.33**	**2.33**	**2.67**	**2**	**2**	**2**	**3**	**2**	**2**	**1.67**	**2**	**2**	**2.03**	**2.75**	**2.25**	**1.75**	**1.5**	**2.06**

3 = Strong Contribution, 2 = Moderate Contribution, 1 = Slight Contribution, --- = No Significant Contribution

Assessment Method (As per the MCA Internship guideline)