MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Autonomous Institution Affiliated to VTU

Competency Based Syllabus (CBS)
for

Master of Computer Applications

(Under Outcome Based Education (OBE) and Choice-Based Credit System (CBCS))

Offered from 1st to 2nd Semesters of Study

In

Partial Fulfillment for the Award of Master's Degree in

Master of Computer Applications

2023 Scheme

Scheme Effective from the academic year 2023-24

General Contents of Competency Based Syllabus Document

Index	Description
1	Prerequisites
2	Competencies
3	Syllabus
4	Syllabus Timeline
5	Teaching-Learning Process Strategies
6	Assessment Details
7	Learning Objectives
8	Course Outcomes and Mapping with POs
9	Assessment Plan
10	Future with this Subject

1st Semester	**BASIC SCIENCE COURSE (BS) MATHEMATICAL FOUNDATION FOR COMPUTER APPLICATIONS**	**M23MCA101**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Combinatorics & Discrete Mathematics**	Basic knowledge of Combinatorics, probability theory and types of functions
2	**Linear algebra**	Familiarity with linear algebra and basic counting methods such as binomial coefficient is assumed
3	**Mathematics**	Proficiency in algebra for Boolean expressions implification using K-map techniques
4	**Fundamental Mathematics Knowledge**	Knowledge of basic algebraic mathematics like union intersections permutations and combinations and binomial Theorem.
5	**Relations and Functions**	Ability to analyze Cartesian product of set and identify the relations
6	**Algebra**	Proficiency in algebraic manipulations, factorization techniques, and solving algebraic equations is necessary for dealing with functions effectively.
7	**Matrices and Determinants**	While not directly related to functions, knowledge of matrices and determinants can be helpful in certain types of function problems.
8	**Probability and Statistics**	Understanding basic probability concepts and statistics can be useful in certain types of function problems that involve probability distributions or data analysis.
9	**Previous Coursework**	Completion of introductory courses in Basic electronics or a related field

2. Competencies

S/L	Competency	KSA Description
1	**Sets, Inclusion and Exclusion Principle, Eigen Values and Eigen Vectors**	**Knowledge:** • Inclusion and Exclusion Principle, Eigen values and Eigen Vectors **Skills:** • Using Eigen values and Eigenvectors in image and signal processing tasks, such as image compression, denoising, and feature extraction. In image **Attitudes:** • Appreciation for the importance of inclusion-exclusion principle whereas maximum flow problem is solved using Ford-Fulkerson algorithm.
2	**Mathematical Logic**	**Knowledge:** Understanding the Basic Connectives, Proof of Theorems **Skills:** • Model Theory, Set Theory, Proof of Theorems **Attitudes:** • Appreciation for analyzing the properties of mathematical structures, and verifying the correctness of computer programs.
3	**Functions and Relation**	**Knowledge:**

		• Understanding of Types of Functions and Relations **Skills:** • Applying Relations and Functions for connection from a row of data to a column or type of data **Attitudes:** • Valuing the importance of evaluate the effectiveness of data structures and algorithms
4	**Random variable and probability distribution**	**Knowledge:** • Understanding the algorithm development, data analysis, machine learning, and simulation modeling. **Skills:** • Applying Probability to analyze data analysis, statistical inference, and machine learning **Attitudes:** • Valuing the importance in decision and estimation problems, and constructs computer algorithms for generating observations from the various distributions.
5	**Graph Theory**	**Knowledge:** • Graphs, Euler Trails and Circuits and Hamilton paths • Knowledge of Graph Coloring and directed graph **Skills:** • Ability to apply graph theory can describe the structure of the circuit by using a directed graph **Attitudes**: • Appreciation for the importance of graph theory in Modeling transportation Network Analysis,

3. Syllabus

MATHEMATICAL FOUNDATION FOR COMPUTER APPLICATIONS **SEMESTER I**			
Course Code	**M23MCA101**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Course objectives: This course will enable students to: 1. To introduce the concepts of mathematical logic. 2. To introduce the concepts of sets, relations, and functions. 3. To perform the operations associated with sets, functions, and relations. 4. To relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context. 5. To use Graph Theory for solving problems.			
Module -1			
Basic Structures: Sets: Principle of Inclusion, Exclusion and Matrices: Eigenvalues and Eigenvectors			L1, L2,L3
Module -2			
Mathematical Logic: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences Predicates and Quantifiers, Nested Quantifiers, Rules of Inference Introduction to Proofs			L1, L2,L3
Module -3			

Functions and Relations: Function, Relations and Their Properties, Pigeonhole principle, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings	L1, L2, L3
Module -4	
Random variable and probability distribution: Concept of random variable, discrete probability distributions, continuous probability distributions, Mean, variance and Co-variance and co-variance of random variables. Binomial and normal distribution, Exponential and normal distribution with mean and variables and problems	L1, L2,L3
Module -5	
Graph Theory: Graphs and Graphs models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring	L1, L2,L3

Text Books

1. Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill publications, 7th edition.
2. Wolpole Myers Ye "Probability and Statistics for engineers and Scientist" Pearson Education, 8th edition.

References Books

1. Richard A Johnson and C.B Gupta "Probability and statistics for engineers" Pearson Education.
2. J.K Sharma "Discrete Mathematics", Mac Millian Publishers India, 3rd edition,2011.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Sets, Inclusion and Exclusion Principle, Eigen Values and Eigen Vectors**	Sets, Worked Problems Principle of Inclusion, Exclusion, Worked Problems Matrices, Worked Problems Eigenvalues and Eigenvectors, Worked Problems
2	**Week 3-4: Mathematical Logic**	Propositional Logic, Applications of Propositional Logic Propositional Equivalences Predicates and Quantifiers, Worked Problems Rules of Inference, Introduction to Proofs, Worked Problems
3	**Week 5-6: Functions and Relations**	Function, Worked Problems Relations and Their Properties Pigeonhole principle, Worked Problems Representing Relations, Closures of Relations, Worked Problems Equivalence Relations, Partial Orderings
4	**Week 7-8: Random variable and probability distribution**	Concept of random variable Discrete probability distributions Continuous Probability distributions, Mean, variance Binomial Distribution, Exponential Distribution, Worked Problems Normal distribution, Worked Problems
5	**Week 9-10: Graph Theory**	Graphs and Graphs models Special Types of Graphs Representing Graphs and Graph Isomorphism, Worked Problems Euler and Hamilton Paths, Shortest-Path Problems Planar Graphs, Graph Coloring
6	**Week 11-12: Integration and Practical Applications**	Apply learned concepts and competencies to real-world scenarios. Hands-on practice

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding the concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Course (PC)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 (TWO) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Set Theory, Graph Theory and Probability**	Students will learn Use Graph theory in modeling transportation networks, including road networks, railway systems, and flight routes. Traffic optimization and resource allocation by analyzing the connectivity and distances between locations within the network.
2	**Designing Huffman Coding**	Students will learn to design the Huffman code with the help of trees, routed tress and Prefix codes

3	**Proficiency in Prefix code**	Students will become proficient in writing Prefix code, Dijkstra's Shortest path algorithm and the algorithms of kruskal and prism
4	**Project-Based Learning**	Through hands-on projects, students will apply their knowledge of Make use Dijkstra's Shortest path algorithm, transport networks..
5	**Collaboration and Communication Skills**	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	**Ethical and Professional Responsibility**	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA101.1	Apply the fundamentals of set theory and matrices for the given problem.
M23MCA101.2	Solve the given problem by applying the Mathematical logic concepts
M23MCA101.3	Identify and list the different applications of discrete mathematical concepts in computer applications
M23MCA101.4	Apply the types of distribution, evaluate the mean and variance for the given case study/ problem
M23MCA101.5	Model the given problem by applying the concepts of graph theory.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA101.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA101.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA101.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA101.4	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA101.5	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA101	**3**	**3**						

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**2**	**2**	**2**	**2**	**2**	**10**
Module 2	**2**	**2**	**2**	**2**	**2**	**10**
Module 3	**2**	**2**	**2**	**2**	**2**	**10**
Module 4	**2**	**2**	**2**	**2**	**2**	**10**
Module 5	**2**	**2**	**2**	**2**	**2**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**4**	**4**	**4**	**4**	**4**	**20**
Module 2	**4**	**4**	**4**	**4**	**4**	**20**
Module 3	**4**	**4**	**4**	**4**	**4**	**20**

Module 4	**4**	**4**	**4**	**4**	**4**	**20**
Module 5	**4**	**4**	**4**	**4**	**4**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

- The "Mathematical Foundation for Computer Application" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions.

- **Cryptography:** The course contributes to the understanding of algorithms using paths in any graph and block encryption algorithms using directed graphs an encryption method in which a graph is the key. Encryption is done by charting a path on that graph. A sequence of vertices in the path of the key graph forms the plain text. A sequence of edges between those vertices forms the cipher text. The girth of a simple graph G is the length of its shortest cycle. Simple graphs of large girth turn out to be useful in networking, error correction theory, Cryptography and other problems of Computer Science.

- **Computer Engineering:** Shortest path algorithms have many applications. As noted earlier, mapping software like Google or Apple maps makes use of shortest path algorithms. They are also important for road network, operations, and logistics research. Shortest path algorithms are also very important for computer networks, like the Internet. Relationships. For linear relationships, as you increase the independent variable by one unit, the mean of the dependent variable always changes by a specific amount. This relationship holds true regardless of where you are in the observation space**.**

1st Semester	**Integrated Professional Core Course (IPC)** **Operating System Concepts**	**M23MCA102**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic understanding of Operating system	Familiarity with fundamental concepts of operating system.
2	System Structure	• Knowledge of single and multiprocessor systems. • Understanding process, memory, storage and protection and security.
3	Process Concepts	Basic understanding of process, scheduling and client server system
4	Memory Management	Understanding of memory and different types and the need of computer memory
5	File System	Basic understanding of files, different permissions given to file and file sharing

2. Competencies

S/L	Competency	KSA Description
1	**Basic understanding of Operating system and system structure**	**Knowledge:** • Understanding of the fundamental concepts of operating systems, including process management, memory management, file systems and input/output operations. • Familiarity with the structure and components of an operating system, including kernels, device drivers, shells and user interfaces. • Knowledge of different types of operating systems such as Windows, Linux, macOS, and their respective features and functionalities. **Skills:** • Ability to navigate and perform basic tasks using various operating system interfaces, including command-line interfaces (CLI) and graphical user interfaces (GUI). • Knowledge of system monitoring tools and techniques to analyze system performance and resource utilization. **Attitudes:** • Ability to adapt to different operating system environments and quickly learn new features and functionalities. • Attention to detail and ability to follow best practices for system configuration, maintenance and security.
2	**Process Concepts**	**Knowledge:** • Understanding of the concept of a process in an operating system, including its definition, attributes, and life cycle stages • Knowledge of process states, transitions between states, and the role of the scheduler in managing process execution. • Familiarity with process control blocks (PCBs) and their contents, including process ID, state, priority, program counter and CPU registers. • Knowledge of inter process communication (IPC) mechanisms, including message passing, shared memory and remote procedure calls, to facilitate communication and coordination between processes. **Skills:** • Skill in implementing process scheduling algorithms, such as round-robin, shortest job first (SJF), and priority scheduling, to allocate CPU resources efficiently. • Skill in analyzing system performance metrics, such as CPU utilization, throughput and response time to optimize process

		scheduling and resource allocation. **Attitudes**: • Critical thinking skills to evaluate different process scheduling algorithms and select the most appropriate one based on system requirements and constraints.
3	**Multi-threading programming**	**Knowledge** • Understanding of the concept of multi-threading and its advantages, including increased responsiveness, improved resource utilization and simplified program structure. • Knowledge of threading models and paradigms, such as user-level threads (ULTs), kernel-level threads (KLTs) and hybrid threading models • Knowledge of multi-threading issues and challenges, such as race conditions, deadlocks and thread starvation. **Skills**: • Proficiency in creating and managing threads using threading libraries and APIs provided by programming languages and operating systems. • Skill in designing and implementing multi-threaded algorithms and data structures improve program performance. **Attitudes**: • Ability to design and implement robust and scalable multi-threaded software systems that meet performance, reliability, and scalability requirements.
4	**Synchronization**	**Knowledge:** • Understanding of the concept of process synchronization and its importance in concurrent programming to ensure data consistency and avoid race conditions. • Familiarity with different synchronization problems and classical synchronization algorithms, such as dining philosophers' problem. **Skills:** • Skill in designing and implementing synchronization protocols to coordinate access to shared resources among multiple threads or processes • Proficiency in applying synchronization techniques to improve the performance, scalability and efficiency of concurrent software systems **Attitudes:** • Critical thinking skills to identify potential synchronization problems and apply appropriate synchronization techniques toaddress them.
5	**Deadlock**	**Knowledge:** • Knowledge of necessary conditions for deadlock, including mutual exclusion, hold and wait, no preemption, and circular wait. • Familiarity with deadlock prevention, avoidance, detection and recovery techniques. **Skills:** • Skill in designing and implementing deadlock prevention and avoid an mechanisms, such as resource ordering and dead lock detection algorithms. • Proficiency in implementing deadlock recovery strategies, such as process termination, resource preemption, and roll back mechanisms. **Attitudes**:

		• Ability to design and implement robust and resilient software system that minimize the occurrence and impact of deadlocks.
6	**Memory Management**	**Knowledge:** • Understanding of memory management concepts in operating systems, including memory allocation, deallocation and protection. • Familiarity with memory addressing modes and techniques such as virtual memory, segmentation, paging and memory-mapped I/O. **Skills:** • Ability to diagnose and troubleshoot memory-related issues such as memory leaks, segmentation faults and out-of-memory errors. • Skill in optimizing memory usage and performance through memory profiling, leak detection and memory foot print analysis. **Attitudes**: • Ability to design and implement robust and scalable software systems that effectively manage memory resources and support dynamic memory allocation and deallocation.
7	**Virtual memory**	**Knowledge**: • Understanding of virtual memory concepts, including address translation, demand paging and page replacement algorithms. • Understanding of the role of the operating system in managing virtual memory, including allocating and deallocating virtual memory space, handling page faults and swapping. **Skills**: • Proficiency in configuring and tuning virtual memory parameters and settings to optimize system performance and resource utilization. • Skill in designing and implementing software systems that efficiently utilize virtual memory resources and minimize overhead. **Attitudes**: • Ability to design and implement robust and scalable software systems that effectively utilize virtual memory to support large-scale applications and workloads.

3. Syllabus

OPERATING SYSTEM CONCEPTS
SEMESTER – I

Course Code	**M23MCA102**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory+10 hours Lab**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course Objectives:

1. It has been expanded to include multi core CPUs, clustered computers and open-source operating systems.
2. It provides significantly updated coverage of virtual machines, as well as multicore CPUs and operating-system debugging.
3. It provides new coverage of pipes as a form of inter process communication.
4. It adds new coverage of programming for multi core systems.
5. It adds a discussion of mutual exclusion locks, priority in version and transactional memory.
6. It updates the Solaris example to include Solaris 10 memory management.

Module -1

OVERVIEW: Introduction to System Structures

What Operating Systems do, Computer-System Organization, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and

Security, Distributed Systems, Special-Purpose Systems, Computing Environments, Open-Source Operating Systems

Operating-System Services, User Operating-System Interface, System calls Types of System calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Virtual Machines Operating-System Debugging.

Module -2

PROCESS MANAGEMENT: Process Concept, Multithreaded Programming

Process concept, Process concept, process scheduling, operations on processes, inter process communication, communication in client-server systems.

Introduction to multithreading, Multithreading models, threading issues, operating-system examples.

Module -3

PROCESS COORDINATION: Synchronization, Deadlocks

Synchronization, deadlocks background, the critical-section problem, Peterson's solution, synchronization hardware, semaphores, classic problems of synchronization, monitors, atomic transactions.

System model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Module -4

MEMORY MANAGEMENT: Memory-Management Strategies, Virtual-Memory Management

Background, swapping, contiguous memory allocation, paging, structure of the page table, segmentation.

Background, demand paging, copy-on-write, page replacement, allocation of frames, thrashing.

Module -5

INTRODUCTION OF UNIX AND LINUX:

Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tput, bc, script, spell and ispell.

Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators &amp; &amp; and ||, exit, if, and case conditions, expr, sleep and wait, while, until, for, $, @, redirection. The here document, set, trap, Sample Validation and Data Entry Scripts.

PRACTICAL COMPONENT OF IPCC

1. Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin(pre-emptive) d) Priority
2. Write a C program to simulate the MVT and MFT memory management techniques.
3. Write a C program to simulate paging technique of memory management.
4. Write a C program to simulate Banker’s algorithm for the purpose of deadlock avoidance.
5. Write a C program to simulate producer-consumer problem using semaphores.
6. Write a C program to simulate the concept to Dining-Philosophers problem.

Text Books:

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating Systems Principles, 8thEdition, Wiley–India.

Reference Books:

1. DM Dhamdhere: Operating Systems–A Concept Based Approach, 2nd Edition, Tata McGraw–Hill,2002.
2. P C P Bhatt: Operating Systems, 2nd edition, PHI,2006.
3. Harvey MDeital: Operating Systems, 3rd edition, Addison Wesley, 1990.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction to Operating Systems	**Competency:** • Understanding the fundamental concepts and functions of operating systems. **Knowledge:** • History and evolution of operating systems.

		• Different types of operating systems and their characteristics. **Skills**: • Explain the purpose and role of an operating system in computing. • Identify and differentiate between various types of operating systems.
2	Week 3-4: Process Management	**Competency**: • Managing processes effectively within an operating system environment. **Knowledge**: • Process life cycle: creation, scheduling, termination. • Various process scheduling algorithms and their implications. **Skills**: • Analyze and implement different process scheduling algorithms. • Demonstrate proficiency in process creation, synchronization and communication.
3	Week 5 -6 Process Coordination	**Competency**: • Understanding the importance of coordinating processes in an operating system. **Knowledge**: • Concept of process coordination and its significance. • Types of process coordination mechanisms. **Skills**: • Explain the need for process coordination in multitasking environments. • Identify different types of process coordination mechanisms
4	Week 7-8: Memory Management	**Competency**: • Efficiently managing memory resources in an operating system. **Knowledge**: • Memory hierarchy and organization. • Virtual memory concepts and techniques. **Skills**: • Implement memory allocation strategies such as paging and segmentation. • Configure and manage virtual memory systems.
5	Week 9-12: Introduction of Unix and Linux	**Competency**: • The ability to effectively navigate, manage, and configure Unix/Linux systems at a basic level. **Knowledge**: • Familiarity with essential Unix/Linux commands for navigation, file manipulation, and system management (e.g., ls, cd, cp, mv, rm, chmod, ps, kill). • Understanding the origins, evolution, and major distributions of Unix and Linux. **Skills**: • The ability to efficiently navigate and operate within the Unix/Linux command-line interface (CLI). • The ability to write and execute simple shell scripts to automate repetitive tasks.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
5	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
6	Laboratory Learning	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

CIE Split up for Integrated Professional Core Course (IPC)

The minimum CIE marks requirement is 50% of maximum marks in each component.

Components		Number	Weightage	Max. Marks	Min. Marks
Theory (A)	Internal Assessment-Tests (A)	2	**60%**	**15**	**7.5**
	Assignments/Quiz/Activity (B)	2	**40%**	**10**	**05**
	Total Marks		**100%**	**25**	**12.5**
Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Laboratory(B)	Record Writing	Continuous	**60%**	15	7.5
	Test at the end of the semester	1	**40%**	10	05
	Total Marks		**100%**	**25**	**12.5**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

1. Question paper pattern will be ten questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding OS Fundamentals	Understanding of the fundamental concepts and functions of operating systems, including process management, memory management, file
2	Process Management	Able to describe the life cycle of a process, analyze different process scheduling algorithms, and implement process synchronization mechanisms to ensure proper coordination among current processes.
3	Problem-solving and Application	Apply the knowledge of operating system concepts and principles to solve practical problems, analyze real-world case studies and design solutions to address specific challenges in OS design and implementations.
4	Advanced OS Concepts	Explore advanced topics in operating systems such as multi-processor and distributed OS concepts, real-time operating systems, virtualization and

		containerization, and evaluate their suitability for different computing environments.
5	Critical Thinking and Reflection	Critically evaluate operating system designs, analyze the impact of emerging technologies on OS development, and reflect on their learning experiences to identify areas for further exploration and improvement.
6	Security and Protection	Identify security threats and vulnerabilities in operating systems, implement authentication and access control mechanisms and configure security features to protect system resources from unauthorized access.
7	Unix and Linux	Understand the directory structure and file hierarchy in Unix/Linux. Write and execute simple shell scripts to automate tasks.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA102.1	Understand the operating system concepts, structure and operations with the system calls
M23MCA102.2	Apply the concepts of operating system such as scheduling, deadlock management, file management and memory management using modern tools
M23MCA102.3	Analyze the process management concepts, threads and their communication and memory management techniques and paging
M23MCA102.4	Evaluate different conditions for dead lock and their possible solutions. Ability to solve synchronization problems.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA102.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA102.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA102.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA102.4	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA102	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	Total
Module 1	10				**10**
Module 2		5	5		**10**
Module 3			5	5	**10**
Module 4		5	5		**10**
Module 5		5	5		**10**
Total	10	15	20	5	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	Total
Module 1	20				**20**
Module 2		10	10		**20**
Module 3			10	10	**20**
Module 4		10	10		**20**
Module 5		10	10		**20**
Total	20	30	40	10	**100**

10. Future with this Subject:

- **Integration with Emerging Technologies:** As technology advances, operating systems will need to integrate with emerging technologies such as artificial intelligence (AI), machine learning (ML), Internet of Things (IoT), edge computing, and quantum computing. Operating systems will evolve to support these technologies and provide efficient resource management, security and interoperability.

- **Enhanced Security and Privacy**: With the increasing threats to cyber security and privacy, future operating systems will focus on enhancing security features such as secure boot, secure enclaves, encryption, and authentication mechanisms. There will also be a greateremphasisonprivacy-preservingtechnologiesanddataprotectionmechanisms.

- **Distributed and Decentralized Systems**: Operating systems will evolve to support distributed and decentralized computing environments, including cloud computing, peer-to-peer networks, and block chain-based systems. This will require advancements in distributed operating systems, resource allocation algorithms, and network protocols.

- **Containerization and Virtualization:** Containerization and virtualization technologies will continue to play a significant role in the future of operating systems, enabling efficient deployment, scaling, and management of applications and services. Operating systems will evolve to provide better support for contain erization platforms such as Docker and Kubernetes, as well as light weight virtualization technologies like micro VMs.

- **Edge Computing and Internet of Things (IOT):** With the proliferation of IOT devices and edge computing infrastructure, operating systems will need to support resource-constrained environments, real-time processing, and low-latency communication. Future operating systems will be optimized for edge computing scenarios and provide support for IOT protocols, device management and data aggregation.

- Overall, the future of Operating System Concepts will be shaped by advancements in technology, changing computing paradigms, and evolving user needs, driving innovation in areas such as security, distributed computing, containerization, edge computing and user experience.

1st Semester	**PROFESSIONAL COURSE (PC) DATA STRUCTURES**	**M23MCA103**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming Knowledge**	Understanding basic programming concepts such as variables, loops, conditionals, and functions. It's usually expected that you know at least one programming language, often C, or C++.
2	**Basic Mathematics**	Familiarity with basic math concepts, particularly discrete mathematics, which includes topics like logic, sets, and functions.
3	**Problem-Solving Skills**	Ability to think logically and solve problems step-by-step.
4	**Basic Algorithms**	Understanding of basic algorithms, such as sorting and searching.
5	**Foundation skills**	Foundational skills needed to understand and work with more complex data structures.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Pointers, Structures and Unions**	**Knowledge:** • Understanding Pointers in C, Structures and Unions **Skills:** • Functions, Call by Value/ Reference. • Recursion, pointers as function arguments **Skill:** • Skill in declaring structures and unions in C programming, understanding the syntax and usage of struct and union keywords. **Attitudes:** • Valuing the importance of pointers in C • Developing a problem-solving orientation towards using structures and unions to address various programming challenges, such as organizing complex data or optimizing memory usage.
2	**Data Structures and Queues**	**Knowledge:** • Understanding of data structures and its various types, understanding of queues. **Skills:** • Representation, operations, applications of queue variants. **Attitudes:** • Appreciation for usage of queues
3	**Stack**	**Knowledge:** • Understanding stacks. **Skills:** • Operations, Applications of stack • Recursion **Attitudes:** • Valuing the importance of recursion
4	**Linked List**	**Knowledge:** • Understanding Linked List **Skills:** • Linked implementations of stacks and queues • Memory management functions **Attitudes:** • Advantages of Linked List over arrays

5	Trees	**Knowledge:** • Understanding of Trees Understanding of hash tables as data structures **Skills:** • Tree Operations • Hashing Techniques **Attitudes:** • Used in various algorithms and data manipulation tasks, including sorting, searching, and traversal.

3. Syllabus

DATA STRUCTURES SEMESTER – I			
Course Code	**M23MCA103**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(4:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**50 hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**
Course objective: • Understand the knowledge of various data structures, operations and algorithms sorting and searching operations. • Implement and analyze the performance of Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques. • Implement all the applications of Data Structures in a high-level language. • Apply appropriate data structures for solving computing problems.			
Module -1			
Introduction to Pointers: Pointers, pointer operations, Pointer Expression, Pointer as function arguments, Functions returning pointers, dynamic memory allocation, array operators using pointers. Pre-processor Directives, Command Line Arguments. **Structures and Unions:** Declaring and using structure and unions with programming examples.			L1, L2, L3
Module -2			
Data Structures: Definition, types: Primitive and Non- Primitive, Linear and Non-linear; Data structure Operations. **Queue:** Definition, array Representation of queues, Operations, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues. Programming Examples.			L1, L2, L3
Module -3			
Stack: Introduction, array Representation of stacks, Operations on stack. **Applications of stack:** Conversion of Arithmetic expressions: Conversion from Infix to postfix, infix to prefix, Evaluation of postfix expression. **Recursion** - Factorial, GCD, Fibonacci numbers, Tower of Hanoi, Binary search, Merge sort.			L1, L2, L3
Module -4			
Linked List: Limitations of array implementation, Memory management functions: Definition, Representation, Operations: getnode() and Freenode() operations, Types: Singly Linked List, Linked list as a data Structure, Inserting and removing nodes from a list, Linked implementations of stacks and queues, Header node linked list, circular linked list, doubly linked list.			L1, L2, L3
Module -5			
Trees: Definitions, Terminologies, Representation of Binary Trees, Types- Complete/full, Almost Complete, Strictly, skewed; Traversal methods - Inorder, postorder, preorder; Binary Search Trees - Creation, Insertion, Deletion, Traversal, Searching in Binary Search Trees. **Hashing**: Hash function, Hash table, collision resolution techniques.			L1, L2, L3
Text Books: 1. Programming in ANSI C, 7th Edition, E. Balagurusamy, McGraw Hill Education. (Chapters 10,11,13,14)			

2. Data Structures by Seymour Lipschutz, Revised First Edition, McGraw Hill Education. (5.6, 7.1-7.9, 9.9)

3. Let us C, Yashwant Kanetkar, BPB Publications

Reference Books:

- Systematic approach to Data Structures using C by A M Padma Reddy, Revised Edition 2009, Sri Nandi Publications, Bangalore.

Skill Development Activities Suggested

The students with the help of the course teacher can take up technical activities which will enhance their skill, or the students should interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to Pointers Structures and Unions**	• Pointers, Structures, Union. • Pointer declaration and accessing, Structure declaration, accessing • Applying Pointers call by value/reference, Implementation of structures and unions
2	**Week 3-4: Data Structures Queues**	• Data Structures- Types. Queues - Types • Knowledge of queue variants • Implementing various types of queues
3	**Week 5-6: Divide and Conquer & Greedy Technique**	• Stacks • Operations, Applications of stack • Implementation of stack
4	**Week 7-8: Space and Time Tradeoffs& Dynamic Programming**	• Recursion • Applications of Recursion • Implementation of recursive programs
5	**Week 9-10: Limitations of Algorithm Power**	• Linked Lists • Understanding of using linked lists to implement other data structures • Performing basic operations on linked lists
6	**Week 1-2: Introduction to Tress and its operations**	• Trees • Understanding of Trees, hashing tables • Tree Operations and Hashing Techniques

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World**	Discuss practical applications to connect theoretical concepts with real-world competencies.

	Application	
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. **Assessment Details (both CIE and SEE)**

CIE Split up for Professional Course (PC)

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. **Learning Objectives**

S/L	**Learning Objectives**	**Description**
1	**Data structures and its operations**	Knowledge of various data structures, operations and algorithms sorting and searching operations
2	**Types of data structures**	Performance of Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques.
3	**Applying data structures**	Appropriate data structures for solving / computing problems.
4	**Implement using programming**	Applications of Data Structures in a high-level language.

8. **Course Outcomes (COs) and Mapping with POs**

Course Outcomes (COs)

COs	**Description**
M23MCA103.1	Understand the concept of pointers, structures and unions and their significance in memory management within programming languages.
M23MCA103.2	Explore and analyze different Data Structures; demonstrate the concept of stack, recursion and queue.
M23MCA103.3	Analyze and apply the concept of Linked list, trees in problem solving.
M23MCA103.4	Implement all data structures in a high-level language for problem solving.

CO-PO Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	-	-	-	-	-	-	-
M23MCA103.1	-	3	-	-	-	-	-	-
M23MCA103.2	3	3	-	-	-	-	-	-
M23MCA103.3	-	-	3	-	-	-	-	-
M23MCA103.4	-	-	-	3	-	-	-	-
M23MCA103	3	3	3	3	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	10				10
Module 2		10			10
Module 3			10		10
Module 4			5	5	10
Module 5			5	5	10
Total	10	10	20	10	50

Semester End Examination (SEE)

	CO1	**CO2**	**CO3**	**CO4**	**Total**
Module 1	20				20
Module 2		20			20
Module 3			20		20
Module 4			10	10	20
Module 5			10	10	20
Total	20	20	40	20	100

10. Future with this Subject

- **Growing Demand:** As technology advances, the demand for professionals proficient in data structures will increase due to the escalating volume and complexity of data.
- **Efficient Data Management:** Skilled individuals will be needed to efficiently organize, store, and retrieve data amidst its growing complexity**.**
- **Effective Utilization of Data Structure:** It will be lifelong learning and remembering that students will have from data structure in IT domain, as it will be utilized in design and implementation of applications and effectively those applications can be built in.
- **Resource allocation:** Any project/application will have enormous amount of requirement and effective resource allocation can be done by using various methods of data structures.
- **Advanced Topics**: More emphasis on advanced data structures and algorithms to handle large-scale data efficiently.
- **Practical Applications**: Integration of real-world applications, such as machine learning, big data, and artificial intelligence, to demonstrate the relevance of data structures.

1st Semester	PROFESSIONAL COURSE (PC) COMPUTER NETWORKS	M23MCA104

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Computer Knowledge**:	Understanding of basic computer operations and components
2	**Basic Programming Skills**	Familiarity with at least one programming language, often used for network programming (e.g., C, C++, Python, Java, Scripts).
3	**Fundamentals of Operating Systems**:	Basic knowledge of operating systems, such as processes, memory management, and file systems.
4	**Basic Mathematics**	Understanding of basic math concepts, especially in areas like binary numbers and probability
5	**Understanding of the Internet**	Basic knowledge of how the internet works, including concepts like IP addresses and web browsing.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction: Data Communications**	**Knowledge:** • Knowledge of data communications fundamentals. • Familiarity with network architectures, topologies, and protocols. **Skills:** • Ability to identify and describe the functions of each layer in a network protocol stack. • Skill in packet analysis and network monitoring. **Attitudes:** • Willingness to learn and adapt to evolving technologies and protocols in the field of data communications and networking.
2	**Physical Layer-1: Analog & Digital Signals**	**Knowledge:** • Understanding of analog and digital signals. • Familiarity with data rate limits imposed by the physical medium and transmission technology. **Skills:** • Ability to differentiate between analog and digital signals, analyze their properties. • Capability to implement line coding. **Attitudes:** • Recognition of the challenges posed by transmission.
3	**Physical Layer-2 and Switching**	**Knowledge:** • Understanding of switching concepts. • Knowledge of Spread Spectrum techniques **Skills:** • Ability to design and configure multiplexing systems • Proficiency in implementing Spread Spectrum techniques **Attitudes:** • Willingness to adapt to different network switching paradigms based on specific requirements
4	**Data Link Layer-1: Error Detection &**	**Knowledge:** • Understanding of error detection and correction mechanisms • Knowledge of block coding techniques

	Correction	**Skills:** • Proficiency in implementing block coding techniques. • Skill in generating and verifying parity bits and checksums **Attitudes:** • Willingness to explore and implement different error detection and correction techniques
5	**Data Link Layer-2: Framing**	**Knowledge:** • Understanding of framing techniques. • Understanding of noiseless communication channels. **Skills:** • Proficiency in implementing framing mechanisms. • Capability to implement error control techniques **Attitudes:** • Understanding the significance of maintaining data confidentiality and integrity in communication systems

3. Syllabus

COMPUTER NETWORKS
SEMESTER – I

Course Code	**M23MCA104**	CIE Marks	**50**
Number of Lecture Hours/Week (L:T:P:S)	**(4:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**50 hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course Objectives:

- Understand the basics of computer networks.
- Knowledge of organization of layered concepts
- Simulation of packets in network communication
- Analysis of Data Link Layer
- Simulation of computer network topology

Module -1	
Introduction: Data Communications, Networks, The Internet, Protocols & Standards, Layered Tasks, The OSI model, Layers in OSI model, TCP/IP Protocol suite, Addressing	L1, L2
Module -2	
Physical Layer-1: Analog & Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital conversion (Only Line coding: Polar, Bipolar and Manchester coding), Analog-to-digital conversion (only PCM), Transmission Modes, Digital-to-analog conversion	L1, L2,
Module -3	
Physical Layer-2 and Switching: Multiplexing, Spread Spectrum, Introduction to switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks	L2, L3
Module -4	
Data Link Layer-1: Error Detection & Correction: Introduction, Block coding, Linear block codes, Cyclic codes, Checksum.	L2, L3
Module -5	
Data Link Layer-2: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy channels, HDLC, PPP (Framing, Transition phases only)	L3

Text Books:

1. Behrouz A. Forouzan,: Data Communication and Networking, 4 th Edition Tata McGraw-Hill, 2006

Reference Books

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

3. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007.
4. Nader F. Mir: Computer and Communication Networks, Pearson Education, 2007.

Skill Development Activities Suggested

The students with the help of the course teacher can take up technical –activities which will enhance their skill or the students should interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction : Data Communications**	• Effectively communicate complex technical concepts related to data communications • Understanding of network protocols and standards governing data communication • Skill in configuring and maintaining DNS servers for domain name resolution.
2	**Week 3-4 Physical Layer-1: Analog & Digital Signals**	• Ability to analyze and evaluate different types of signals • Knowledge of common transmission impairments • Proficiency in implementing line coding schemes
3	**Week 5-6: Physical Layer-2 and Switching**	• Ability to design efficient multiplexing and switching systems. • Understanding of the transition phases in switching. • Proficiency in implementing multiplexing techniques.
4	**Week 7-8: Data Link Layer-1: Error Detection.**	• To analyze and resolve errors in data transmission using various error detection techniques. • Familiarity with the principles of checksum algorithms for error detection. • Skill in generating and verifying checksums to detect errors in transmitted data.
5	**Week 9-10: Error Correction**	• To analyze and resolve errors in data transmission using various error correction techniques. • Understanding of how error correction codes work, including Hamming codes and Reed-Solomon codes. • Competence in implementing error correction codes
6	**Week 11-12: Data Link Layer-2: Framing**	• Ability to design efficient framing, flow and error control mechanisms. • Knowledge of flow control mechanisms • Ability to troubleshoot and debug framing, flow, and error control issues.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation /Simulation**	Incorporate visual aids like videos/animations/simulation to enhance understanding of basic concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.

5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Course (PC)

	Components	Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	30	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	20	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Basics of Computer Networks**	Computer networks are essentially a system of interconnected computers and other devices that can communicate with each other. They enable the sharing of resources and information between devices, facilitating tasks ranging from simple file sharing to complex data processing
2	**Organization of Layers**	The organization of layers in computer networks follows the OSI (Open Systems Interconnection) model, which is a conceptual framework for understanding how different networking protocols and technologies interact. The OSI model consists of seven layers, each responsible for specific functions in the communication process.
3	**Packets Communication**	Packet communication is a fundamental concept in computer networking, enabling the transmission of data across networks.
4	**Data Link Layer**	The Data Link Layer, the second layer in the OSI (Open Systems Interconnection) model, plays a crucial role in facilitating node-to-node communication within the same network
5	**Network Topology**	Network topology refers to the physical or logical layout of interconnected devices and nodes in a computer network. It determines how devices are connected, how data flows between them, and the overall structure of the network.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA104.1	Apply the basic concepts of networks like protocol, internet and OSI layers
M23MCA104.2	Analyze the Physical Layer of 1 and 2
M23MCA104.3	Demonstrate the various Switching networks
M23MCA104.4	Analyze the Data Link Layer of 1 and 2

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA104.1	3	-	-	-	-	-	-	-
M23MCA104.2	-	3	-	-	-	-	-	-
M23MCA104.3	-	-	3	-	-	-	-	-
M23MCA104.4	-	3	-	-	-	-	-	-
M23MCA104	3	3	3	-	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	Total
Module 1	**05**	**05**	**-**	**-**	**10**
Module 2	**-**	**05**	**05**	**-**	**10**
Module 3	**-**	**05**	**05**	**-**	**10**
Module 4	**-**	**-**	**05**	**05**	**10**
Module 5	**-**	**05**	**-**	**05**	**10**
Total	**05**	**20**	**15**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	Total
Module 1	20	-	-	-	20
Module 2	-	20	-	-	20
Module 3	-	20	-	-	20
Module 4	-	-	10	10	20
Module 5	-	10	-	10	20
	20	50	10	20	100

10. Future with this Subject

- **Cyber security**: Increased focus on network security, covering topics such as encryption, intrusion detection, and secure communication.
- **Cloud Computing**: Integration of cloud networking concepts, including virtual networks and cloud service models.
- **Real-World Applications**: Use of practical, real-world scenarios to illustrate network design and troubleshooting.
- **Hands-On Learning**: More interactive and hands-on labs using simulation tools and real networking equipment.

1st Semester	**PROFESSIONAL COURSE (PC)** **DESIGN AND ANALYSIS OF ALGORITHMS**	**M23MCA105**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Skills**	• Understanding of basic programming concepts. • Familiarity with fundamental programming concepts such as variables, loops, conditional statements, functions, and data structures. Commonly used languages include Python, Java, C++, and others.
2	**Discrete Mathematics**	• Understanding of fundamental concepts in discrete mathematics is crucial as many algorithmic principles are rooted in discrete structures. • Topics such as sets, relations, functions, logic, combinatorial, and graph theory provide the theoretical basis for algorithm design and analysis.
3	**Data Structures**	• Familiarity with fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs is necessary. • Students should understand how these data structures are implemented and their associated operations and properties. • Proficiency in manipulating and traversing data structures is essential for algorithm design.
4	**Complexity Analysis**	• Basic knowledge of mathematical analysis and reasoning is required to understand the fundamentals of algorithmic complexity. • Students should be familiar with asymptotic notation (Big O, Big Omega, Big Theta) and have the ability to analyze the time and space complexity of algorithms.
5	**Basic Algorithms**	• Exposure to basic algorithmic concepts and techniques is helpful. • Students should understand commonly used algorithms such as sorting, searching, and graph traversal algorithms.
6	**Mathematical Reasoning**	• Proficiency in mathematical reasoning and problem-solving is beneficial. • Students should be able to formulate problems mathematically, identify patterns, and devise strategies for solving them. • Skills in proof techniques and mathematical induction are particularly useful for algorithm analysis.
7	**Logic and Reasoning**	• Strong logical reasoning skills are essential for algorithm design and analysis. • Students should be able to think critically, analyze problem requirements, and devise algorithmic solutions systematically. • Logical reasoning skills are essential for understanding algorithm correctness and complexity.

2. Competencies

S/L	Competency	KSA Description
1	**Fundamentals of design and Analysis of Algorithm**	**Knowledge:** • Understanding of algorithms designing principles. • Knowledge of algorithms analysis for recursive and non-recursive algorithms. **Skills:** • Generating a function which bounds the algorithm's computing time (a priori analysis). • Using asymptotic notation to determine the order of magnitude of the frequency of execution of statements. **Attitudes:** • Appreciation for the importance of of design and Analysis of Algorithm.

2	**Brute Force & Divide and Conquer**	**Knowledge:** • Understanding of Brute Force and Divide-and-Conquer methods **Skills**: • Implementing ordered or linear lists stacks and queues. • Implementing trees: B-Trees, binary trees, heaps. • Designing and implementing solutions using graphs. **Attitudes:** • Appreciation for the importance of Brute Force and Divide-and-Conquer methods.
3	**Decrease-and-Conquer & Greedy Technique**	**Knowledge:** • Understanding the Decrease-and-Conquer & Greedy Technique. **Skills:** • Applying the solution to solve complex problems including the knapsack and job scheduling problems. • Designing and implementing an optimal merge pattern that will reduce the number of operations when merging records. • Applying binary trees with minimal weighted external path lengths to obtain an optimal set of codes for messages. • Developing minimum spanning trees used in graph traversal. **Attitudes:** Recognizing the significance of Decrease-and-Conquer and greedy method**.**
4	**Dynamic Programming method**	**Knowledge:** • Understanding of dynamic programming approach. **Skills:** • Developing a dynamic programming formulation for a k-stage graph problem. • Developing and implementing optimal binary search trees. • Apply dynamic programming algorithms to solve the 0/1 knapsack problem. • Find the minimum cost path to solve the traveling salesperson problem. **Attitudes:** • Openness to analyzing and designing of dynamic programming approaches.
5	**Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems**	**Knowledge:** • Understanding of NP-Hard and NP-complete problems. **Skills:** • Defining what types of problems are NP Hard or NP-complete problems. • Describing the characteristics of approximation algorithms for NP-hard problems. **Attitudes:** • Appreciation for the versatility of NP-Hard and NP-complete problems.
6	**Coping with Limitations of Algorithm Power**	**Knowledge:** • Understanding of back tracking and branch & bound designing techniques. **Skills:** • Creating a tree structure that defines the problems state space of the problem. • Systematically generating the problem states, determining which solution states are, and which solution states are answer states. • Implementing a depth first node and breadth first node generation with bounding functions. • Developing a systematic enumeration of candidate solutions by means of

		state space search. • Enumerating the candidate solutions of a branch by checking against upper and lower estimated bounds on the optimal solution. • Applying the B&B method to the solution of the zero-one knapsack and traveling salesman problems. **Attitudes:** • Appreciation for the role of clear and well-structured back tracking and branch & bound designing techniques.

3. Syllabus

<table>
<tr><td colspan="4"><b>DESIGN AND ANALYSIS OF ALGORITHMS</b>
<b>SEMESTER – I</b></td></tr>
<tr><td>Course Code</td><td><b>M23MCA105</b></td><td>CIE Marks</td><td><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week(L:T:P:S)</td><td><b>(4:0:0:0)</b></td><td>SEE Marks</td><td><b>50</b></td></tr>
<tr><td>Total Number of Lecture Hours</td><td><b>50 hours</b></td><td>Total Marks</td><td><b>100</b></td></tr>
<tr><td>Credits</td><td><b>04</b></td><td>Exam Hours</td><td><b>03</b></td></tr>
<tr><td colspan="4"><b>Course Objectives:</b>
• To impart the concepts of notion of Algorithms and Problem Solving steps, Mathematical analysis of Recursive and Non-recursive algorithms.
• To impart the concepts of designing an efficient algorithm not only limited in reducing cost and time but to enhance scalability, reliability and availability.
• To impart the concept of various Algorithm Designing techniques on various problems.
• To impart the concepts of on the limitations of algorithmic power and how this limitation can be coped Up by using design techniques like backtracking and branch-and-bound, and finally Concludes with a discussion of few approximation algorithms.</td></tr>
<tr><td colspan="4"><b>Module -1</b></td></tr>
<tr><td colspan="3"><b>Introduction:</b> Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures. <b>Fundamentals of the Analysis of Algorithm Efficiency</b>: Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical analysis of Recursive and Non-recursive algorithms.</td><td>L1, L2, L3</td></tr>
<tr><td colspan="4"><b>Module -2</b></td></tr>
<tr><td colspan="3"><b>Brute Force:</b> Selection sort and Bubble sort, Sequential Search and String Matching, Exhaustive search. <b>Divide-and-Conquer:</b> Divide & conquer method, Merge sort, Quick sort, Binary Search, Multiplication of large integers.</td><td>L1, L2, L3</td></tr>
<tr><td colspan="4"><b>Module -3</b></td></tr>
<tr><td colspan="3"><b>Decrease-and-Conquer:</b> Variations in Decrease & conquer method, Insertion Sort, Depth First Search and Breadth First Search, Topological sorting. <b>Greedy Technique:</b> Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffmann Trees.</td><td>L1, L2, L3</td></tr>
<tr><td colspan="4"><b>Module -4</b></td></tr>
<tr><td colspan="3"><b>Space and Time Tradeoffs:</b> Sorting by Counting, Input Enhancement in String Matching. <b>Dynamic Programming:</b> Computing a binomial coefficient, Warshall's and Floyd's Algorithms, the Knapsack Problem and Memory Functions.</td><td>L1, L2, L3</td></tr>
<tr><td colspan="4"><b>Module -5</b></td></tr>
<tr><td colspan="3"><b>Limitations of Algorithm Power:</b> Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems. <b>Coping with Limitations of Algorithm Power:</b> Backtracking: n-Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem. Branch-and-Bound: Assignment Problem, Knapsack Problem, Traveling Salesperson Problem.</td><td>L1, L2, L3</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"><b>Text Books:</b>
• AnanyLevitin: <b>Introduction to the Design and Analysis of Algorithms</b>, Pearson Education, 2nd Edition.(Chapters 1.1-1.4, 2.1-2.4, 3.1, 3.2, 3.4, 4.1-4.5, 5.1-5.4, 7.1-7.3, 8.1, 8.2, 8.4, 9.1-9.4,</td></tr>
</table>

11.1-11.3, 12.1-12.2)

Reference Books:

1. Coremen T.H., Leiserson C.E., and Rivest R.L.: Introduction to Algorithms, PHI 1998.
2. Horowitz E., Sahani S., Rajasekharan S.: Computer Algorithms, Galgotia Publication 2001.
3. Michael T Goodrich and Roberto Tamassia : Algorithm Design, Wiley India R C T Lee, S S Tseng, R C Chang, Y T Tsai : Introduction to Design and Analysis of Algorithms: A Strategic Approach, Tata McGraw Hill

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to algorithms & Fundamentals of the Analysis of Algorithm Efficiency**	• Fundamentals of design and Analysis of Algorithm. • Notion of algorithm, Algorithmic Problem Solving Technique, Problem Types and Data Structures • Analysis Framework, Asymptotic Notations • Applying problem solving techniques to design algorithms • Applying steps for analyzing the recursive and non-recursive algorithms
2	**Week 3-4: Brute Force & Divide-and-Conquer**	• Design of algorithms with Brute Force & Divide and Conquer • Understanding searching, Exhaustive search, sorting, string matching process, multiplication of large integers. • Brute Force and Divide-and-Conquer problem solving Technique to solve the problems • Apply Brute Force to Design and analyze the algorithms Linear Searching, Bubble and Selection Sorting, String matching process. • Apply Divide-and-Conquer Design and analyze the algorithms binary search, merge sort, Quick sort.
3	**Week 5-6: Decrease and Conquer & Greedy Technique**	• Design of algorithms with Decrease-and-Conquer & Greedy Technique. • Variations in Decrease & conquer method, Sort, Depth First Search and Breadth First Search, Topological sorting. • Optimization problems single-source shortest path, MST, Huffmann Trees, • Apply **Decrease-and-Conquer** to Design and analyze the algorithms insertion sort, BFS, DFS. • Apply **Greedy Technique** Design and analyze the algorithms Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffmann Trees
4	**Week 7-8: Space and Time Tradeoffs& Dynamic Programming**	• Design of algorithms with Dynamic Programming method. • Sorting by Counting, Input Enhancement in String Matching. • Computing a binomial coefficient, shortest path problem, the Knapsack Problem and Memory Functions. • Applying Input Enhancement on strings matching process, sorting. • Applying Dynamic Programming to design and analysis on Warshall's and Floyd's Algorithms ,Computing a binomial coefficient, Knapsack Problem and Memory Functions
5	**Week 9-10: Limitations of Algorithm Power**	• Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems • Lower-Bound Arguments of P, NP and NP-Complete Problems. • Writing Decision Trees, implementing P, NP and NP-Complete Problems.

6	**Week 11-12: Coping with Limitations of Algorithm Power**	• Coping with Limitations of Algorithm Power. • Coping with Limitations of Algorithm Power with N-Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem, Assignment Problem, Knapsack Problem, Traveling Salesperson. • Apply Backtracking on n-Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem. • Apply Branch-and-Bound on Assignment Problem, Knapsack Problem, And Traveling Salesperson Problem.

4. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Course (PC)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding of Algorithm Design Paradigms**	Students should be able to comprehend and apply various algorithm design paradigms such as divide and conquer, dynamic programming, greedy algorithms, and others.
2	**Problem-Solving Skills**	Develop problem-solving skills by being able to identify, formulate, and solve algorithmic problems efficiently using appropriate techniques.
3	**Proficiency in Algorithm Analysis**	Gain proficiency in analyzing the time and space complexity of algorithms, including asymptotic notation (Big O, Big Omega, Big Theta), worst-case, average-case, and best-case analysis.
4	**Algorithmic Thinking**	Cultivate algorithmic thinking, which involves breaking down problems into smaller, manageable components, identifying patterns, and devising algorithmic solutions.
5	**Understanding of Data Structures**	Understand the relationship between algorithms and data structures, and be able to select appropriate data structures to optimize algorithmic performance.
6	**Algorithm Implementation**	Be able to implement algorithms in a programming language of choice, translating theoretical knowledge into practical code.
7	**Algorithmic Paradigm Selection**	Develop the ability to select the most suitable algorithmic paradigm for solving a given problem based on its characteristics and constraints.
8	**Ethical and Professional Responsibility**	Students will understand the ethical and professional responsibilities associated with designing algorithms, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.
9	**Critical Thinking and Creativity**	Foster critical thinking and creativity by encouraging students to devise novel algorithmic solutions to complex problems.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA105.1	**Comprehend** the steps for Analyzing the performance of recursive and non-recursive algorithms and use of asymptotic notations to measure the performance of algorithms.
M23MCA105.2	**Apply** prior knowledge of mathematics and standard algorithm design techniques to solve given problems.
M23MCA105.3	**Analyze** the complexities of various problems in different domains and infer the results.
M23MCA105.4	**Design** an algorithm to solve a given problem under various domains.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA105.1	3	-	-	-	-	-	-	-
M23MCA105.2	3	-	-	-	-	-	-	-
M23MCA105.3	-	3	-	-	-	-	-	-
M23MCA105.4	-	-	3	-	-	-	-	-
M23MCA105	3	3	3	-	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**	**-**	**-**	**-**	**10**
Module 2	**-**	**-**	**-**	**-**	**-**
Module 3	**-**	**5**	**10**	**-**	**15**
Module 4	**-**	**5**	**10**	**5**	**20**
Module 5	**-**	**-**	**-**	**5**	**5**
Total	**10**	**10**	**20**	**10**	**50**

Semester End Examination (SEE)

	CO1	**CO2**	**CO3**	**CO4**	**Total**
Module 1	**10**				**10**
Module 2	**10**				**10**
Module 3		**15**	**15**		**30**
Module 4		**10**	**15**	**15**	**40**
Module 5				**10**	**10**
Total	**20**	**25**	**30**	**25**	**100**

10. Future with this Subject

- **Algorithmic Efficiency and Scalability:** With the increasing volume of data generated by various sources such as IoT devices, social media, and sensors, there's a growing demand for algorithms that can efficiently process and analyze large datasets in a scalable manner. Future algorithms will need to be optimized for performance and resource utilization, taking advantage of parallel processing, distributed computing, and advancements in hardware architectures like GPUs and TPUs.
- **Machine Learning and AI Integration:** Machine learning and artificial intelligence techniques are increasingly being integrated into algorithm design and analysis. Algorithms that can learn and adapt to changing data patterns, optimize themselves over time, and make decisions autonomously will become more prevalent. This integration will lead to the development of hybrid algorithms that combine traditional algorithmic approaches with machine learning models.
- **Quantum Computing:** The advent of quantum computing has the potential to revolutionize algorithm design and analysis. Quantum algorithms can solve certain types of problems exponentially faster than classical algorithms, particularly in areas such as cryptography, optimization, and simulation. As quantum computing technology matures, there will be a need for algorithms that exploit its unique properties while also addressing challenges such as noise and error correction.
- **Algorithmic Fairness and Ethics:** There is a growing awareness of the social and ethical implications of algorithms, particularly in areas like bias, privacy, and transparency. Future algorithms will need to be designed and analyzed with a greater emphasis on fairness, accountability, and transparency. This may involve incorporating ethical considerations into the algorithm design process, developing techniques for detecting and mitigating bias, and ensuring that algorithms are interpretable and explainable.
- **Interdisciplinary Approaches:** The boundaries between different fields such as computer science, mathematics, and domain-specific areas are becoming increasingly blurred. Future advancements in algorithm design and analysis are likely to emerge from interdisciplinary collaborations, where insights from diverse fields are combined to tackle complex problems. This interdisciplinary approach may lead to the development of algorithms that are tailored to specific application domains, such as healthcare, finance, or environmental science.

- Overall, the future of algorithm design and analysis is likely to be characterized by a combination of advancements in computational techniques, integration with emerging technologies, and a greater emphasis on ethical and societal considerations.

In summary, the “Design and analysis of Algorithms" course serves as a stepping stone, equipping students with foundational knowledge and skills that are essential for the subsequent courses in their MCA program and for their future careers in various technology-related fields.

1st Semester	**PROFESSIONAL COURSE LABORATORY (PCL)** **DATA STRUCTURES LABORATORY**	**M23MCAL106**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic Programming Knowledge**	Understanding basic programming concepts such as variables, loops, conditionals, and functions. It's usually expected that you know at least one programming language, often C, or C++.
2.	**Basic Mathematics**	Familiarity with basic math concepts, particularly discrete mathematics, which includes topics like logic, sets, and functions.
3.	**Problem-Solving Skills**	Ability to think logically and solve problems step-by-step.
4.	**Basic Algorithms**	Understanding of basic algorithms, such as sorting and searching.
5.	**Foundation skills**	Foundational skills needed to understand and work with more complex data structures.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Pointers, Structures and Unions**	**Knowledge:** • Understanding Pointers in C, Structures and Unions **Skills:** • Functions, Call by Value/ Reference. • Recursion, pointers as function arguments • Skill in declaring structures and unions in C programming, understanding the syntax and usage of struct and union keywords. **Attitudes:** • Valuing the importance of pointers in C • Developing a problem-solving orientation towards using structures and unions to address various programming challenges, such as organizing complex data or optimizing memory usage.
2	**Data Structures and Queues**	**Knowledge:** • Understanding of data structures and its various types, understanding of queues. **Skills:** • Representation, operations, applications of queue variants. **Attitudes:** • Appreciation for usage of queues
3	**Stack**	**Knowledge:** • Understanding stacks. **Skills:** • Operations, Applications of stack • Recursion **Attitudes:** • Valuing the importance of recursion
4	**Linked List**	**Knowledge:** • Understanding Linked List **Skills:** • Linked implementations of stacks and queues • Memory management functions **Attitudes:** • Advantages of Linked List over arrays

5	**Trees**	**Knowledge:** • Understanding of Trees • Understanding of hash tables as data structures **Skills:** • Tree Operations • Hashing Techniques **Attitudes:** • Used in various algorithms and data manipulation tasks, including sorting, searching, and traversal.

3. Syllabus

DATA STRUCTURES LABORATORY
SEMESTER – I

Course Code	**M23MCAL106**	CIE Marks	**50**
Number of Lecture Hours/Week(L:T:P:S)	**(0:0:3:0)**	SEE Marks	**50**
Credits	**02**	Exam Hours	**03**

Course objectives:

- Evaluate the Expressions like postfix, prefix conversions.
- Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs

Sl. No	Experiments
1.	Design a structure time with 3 integer members hours, minutes and seconds using time structure. Write a C program to have 4 variables T1, T2, T3 and T4. Program should include functions to input the time data, to print hh:mm:ss and to add two-time data. Use these functions to find T1+T2+T3+T4.
2.	Write a C program to implement Binary Search using Recursion.
3.	Design, develop, and execute a programming in C to simulate the working of a priority queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display.
4.	Design, develop and execute a program in C to simulate the working of a circular queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
5.	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are +(add), -(subtract), *(multiply) and /(divide).
6.	Write a C program for converting infix expression to postfix expressions.
7.	Write a C program to perform Dequeue operation using Singly Linked List.
8.	Write a C program that implements a singly linked list in ascending order:
9.	Write a C program to insert a node into a Binary Search Tree.
10.	Write a C program that creates an expression tree from a given postfix expression.
11.	Write aC program that demonstrates tree traversal using recursion.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2:** **Implement data Structures and Its functions**	Applying Pointers call by value/reference, Implementation of structures and unions
2	**Week 3-4:** **Implementation of Arrays Queues**	Implementing various types of arrays and queues

3	**Week 5-6: Implementation of stacks and linked list**	Implementation of stack and linked list
4	**Week 7-8: Implement recursion and searching techniques**	Implementation of recursive programs and binary search programs

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE for Practical Courses (Laboratory Based):

- CIE marks for a practical course shall be 50 marks.
- The split up of CIE marks for record/journal and test to be split in the ratio 60:40
- Record write up for individual program/experiment will be evaluated for 10 Marks
- Total marks scored for record writing and conduction shall be scaled downed to 30 marks **(60% of the CIE Lab Marks (50))**
- 1 (one) test for 100 marks after the completion of the experiments at the end of the semester. The Test marks should be scaled down to 20 marks **(40% of the CIE Lab Marks (50))**

Test

Marks distribution for Laboratory based Practical Course for TEST

Sl. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	60
2	Viva-Voce	40%	40
Total		**100%**	**100**

Final CIE in Practical Course:

Marks distribution for Laboratory based Practical Course for Final CIE

Sl. No.	Description	% of Marks	In Marks
1	Scaled Down marks of Record	60% of the maximum	30
2	Scaled Down marks of Test	40% of the maximum	20
Total		**100%**	**50**

SEE for Practical Course (Laboratory based):

SEE marks for practical course shall be 50 marks

Marks distribution for Laboratory based Practical Course for Final SEE

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

See for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.

Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.

Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Data Structures and operations**	Knowledge of various data structures, operations and algorithms sorting and searching operations
2	**Types of data structures**	Performance of Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques.
3	**Applying data structures**	Appropriate data structures for solving/computing problems.
4	**Implement data structures using programming**	Applications of Data Structures in a high-level language.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCAL106.1	Understand the concept of pointers, structures and unions and their significance in memory management within programming languages.
M23MCAL106.2	Explore and analyze different Data Structures; demonstrate the concept of stack, recursion and queue.
M23MCAL106.3	Analyze and apply the concept of Linked list, trees in problem solving.
M23MCAL106.4	Implement all data structures in a high-level language for problem solving.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCAL106.1	3	-	-	-	-	-	-	-
M23MCAL106.2	-	3	-	-	-	-	-	-
M23MCAL106.3	3	3	-	-	-	-	-	-
M23MCAL106.4	-	-	3	-	-	-	-	-
M23MCAL106	3	3	3	-	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Lab Programs	10	-	-	-	**10**
	-	10	-	-	**10**
	-	-	15	-	**15**
	-	-	-	15	**15**
Total	**10**	**10**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	**CO2**	**CO3**	**CO4**	**Total**
Lab Programs	20	-	-	-	20

	-	20	-	-	20
	-	-	30	-	30
	-	-	-	30	30
Total	20	20	30	30	100

10. Future with this Subject

- **Growing Demand:** As technology advances, the demand for professionals proficient in data structures will increase due to the escalating volume and complexity of data.
- **Efficient Data Management:** Skilled individuals will be needed to efficiently organize, store, and retrieve data amidst its growing complexity**.**
- **Effective Utilization of Data Structure:** It will be lifelong learning and remembering that students will have from data structure in IT domain, as it will be utilized in design and implementation of applications and effectively those applications can be built in.
- **Resource allocation:** Any project/application will have enormous amount of requirement and effective resource allocation can be done by using various methods of data structures.
- **Advanced Topics**: More emphasis on advanced data structures and algorithms to handle large-scale data efficiently.
- **Practical Applications**: Integration of real-world applications, such as machine learning, big data, and artificial intelligence, to demonstrate the relevance of data structures.
- **Interdisciplinary Approaches**: Combining data structures with other fields like bioinformatics, cybersecurity, and finance.
- **Online and Interactive Learning**: Increased availability of online courses and interactive tools to enhance learning experiences.
- **Continuous Updates**: Regular updates to the curriculum to include the latest advancements and industry trends

1st Semester	PROFESSIONAL COURSE LABORATORY (PCL) COMPUTER NETWORKS LABORATORY	M23MCAL107

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Computer Knowledge**:	Understanding of basic computer operations and components
2	**Basic Programming Skills**	Familiarity with at least one programming language, often used for network programming (e.g., C, C++, Scripts)
3	**Fundamentals of Operating Systems**:	Basic knowledge of operating systems, such as processes, memory management, and file systems.
4	**Basic Mathematics**	Understanding of basic math concepts, especially in areas like binary numbers and probability
5	**Understanding of the Internet**	Basic knowledge of how the internet works, including concepts like IP addresses and web browsing.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction: Data Communications**	**Knowledge:** • Knowledge of data communications fundamentals. • Familiarity with network architectures, topologies, and protocols. **Skills:** • Ability to identify and describe the functions of each layer in a network protocol stack. • Skill in packet analysis and network monitoring. **Attitudes:** • Willingness to learn and adapt to evolving technologies and protocols in the field of data communications and networking.
2	**Physical Layer-1: Analog & Digital Signals**	**Knowledge:** • Understanding of analog and digital signals. • Familiarity with data rate limits imposed by the physical medium and transmission technology. **Skills:** • Ability to differentiate between analog and digital signals, analyze their properties. • Capability to implement line coding. **Attitudes:** • Recognition of the challenges posed by transmission.
3	**Physical Layer-2 and Switching**	**Knowledge:** • Understanding of switching concepts. • Knowledge of Spread Spectrum techniques **Skills:** • Ability to design and configure multiplexing systems • Proficiency in implementing Spread Spectrum techniques **Attitudes:** • Willingness to adapt to different network switching paradigms based on specific requirements
4	**Data Link Layer-1: Error**	**Knowledge:** • Understanding of error detection and correction mechanisms

	Detection & Correction	• Knowledge of block coding techniques **Skills:** • Proficiency in implementing block coding techniques. • Skill in generating and verifying parity bits and checksums **Attitudes:** • Willingness to explore and implement different error detection and correction techniques
5	**Data Link Layer-2: Framing**	**Knowledge:** • Understanding of framing techniques. • Understanding of noiseless communication channels. **Skills:** • Proficiency in implementing framing mechanisms. • Capability to implement error control techniques **Attitudes** • Understanding the significance of maintaining data confidentiality and integrity in communication systems

3. **Syllabus**

COMPUTER NETWORKS LABORATORY **SEMESTER – I**			
Course Code	**M23MCAL107**	CIE Marks	**50**
Number of Lecture Hours/Week(L:T:P:S)	**(0:0:3:0)**	SEE Marks	**50**
Credits	**02**	Exam Hours	**03**

Course objectives:

- Understand the basics of computer networks.
- Knowledge of organization of layered concepts
- Simulation of packets in network communication

Sl. No	**EXPERIMENTS**
1	Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped
2	Implement the data link layer framing methods such as character, character-stuffing and bit stuffing
3	Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
4	Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
5	Implement Dijsktra's algorithm to compute the shortest path through a network
6	Implement data encryption and data decryption
7	Simulate the network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.
8	Simulate to study transmission of packets over Ethernet LAN and determine the number of packets drop destination.
9	Simulate the different types of internet traffic such as FTP and TELNET over a wired network and analyze the packet drop and packet delivery ratio in the network.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction : Data Communications**	• Effectively communicate complex technical concepts related to data communications • Understanding of network protocols and standards governing data communication • Skill in configuring and maintaining DNS servers for domain name resolution.
2	**Week 3-4 Physical Layer-1: Analog & Digital Signals**	• Ability to analyze and evaluate different types of signals • Knowledge of common transmission impairments • Proficiency in implementing line coding schemes
3	**Week 5-6: Physical Layer-2 and Switching**	• Ability to design efficient multiplexing and switching systems. • Understanding of the transition phases in switching. • Proficiency in implementing multiplexing techniques.
4	**Week 7-8: Data Link Layer-1: Error Detection.**	• To analyze and resolve errors in data transmission using various error detection techniques. • Familiarity with the principles of checksum algorithms for error detection. • Skill in generating and verifying checksums to detect errors in transmitted data.
5	**Week 9-10: Error Correction**	• To analyze and resolve errors in data transmission using various error correction techniques. • Understanding of how error correction codes work, including Hamming codes and Reed-Solomon codes. • Competence in implementing error correction codes
6	**Week 11-12: Data Link Layer-2: Framing**	• Ability to design efficient framing, flow and error control mechanisms. • Knowledge of flow control mechanisms • Ability to troubleshoot and debug framing, flow, and error control issues.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation /Simulation**	Incorporate visual aids like videos/animations/simulation to enhance understanding of basic concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE for Practical Courses (Laboratory Based):

- CIE marks for a practical course shall be 50 marks.
- The split up of CIE marks for record/journal and test to be split in the ratio 60:40
- Record write up for individual program/experiment will be evaluated for 10 Marks
- Total marks scored for record writing and conduction shall be scaled downed to 30 marks **(60% of the CIE Lab Marks (50))**
- 1 (one) test for 100 marks after the completion of the experiments at the end of the semester. The Test marks should be scaled down to 20marks **(40% of the CIE Lab Marks (50)) Test**

Marks distribution for Laboratory based Practical Course for TEST

Sl. No.	**Description**	**% of Marks**	**In Marks**
1	Write-up, Conduction, result and Procedure	60%	60
2	Viva-Voce	40%	40
Total		**100%**	**100**

Final CIE in Practical Course:

Marks distribution for Laboratory based Practical Course for Final CIE

Sl. No.	**Description**	**% of Marks**	**In Marks**
1	Scaled Down marks of Record	60% of the maximum	30
2	Scaled Down marks of Test	40% of the maximum	20
Total		**100%**	**50**

SEE for Practical Course (Laboratory based):

Marks distribution for Laboratory based Practical Course for Final SEE

SL. No.	**Description**	**% of Marks**	**Marks**
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

1. SEE marks for practical course shall be 50 marks
2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	**Learning Objectives**	**Description**
1	**Basics of Computer Networks**	Computer networks are essentially a system of interconnected computers and other devices that can communicate with each other. They enable the sharing of resources and information between devices, facilitating tasks ranging from simple file sharing to complex data processing
2	**Organization of Layers**	The organization of layers in computer networks follows the OSI (Open Systems Interconnection) model, which is a conceptual framework for understanding how different networking protocols and technologies interact. The OSI model consists of seven layers, each responsible for specific functions in the communication process.
3	**Packets Communication**	Packet communication is a fundamental concept in computer networking, enabling the transmission of data across networks.
4	**Data Link Layer**	The Data Link Layer, the second layer in the OSI (Open Systems Interconnection) model, plays a crucial role in facilitating node-to-node communication within the same network

5	**Network Topology**	Network topology refers to the physical or logical layout of interconnected devices and nodes in a computer network. It determines how devices are connected, how data flows between them, and the overall structure of the network.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	**Description**
M23MCAL107.1	Apply suitable methodology for building familiar network and associated algorithms with C/C++ and TCL scripting language.
M23MCAL107.2	Analyze given problem scenario, infer the corrections of the selected parameters based on efficiency of solution and document the same.
M23MCAL107.3	Design network topology with different protocols for better performance using NS2
M23MCAL107.4	Conduct experiments either individually or in a team and present its corresponding outcomes and process both orally and in written form.

CO-PO Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
M23MCAL107.1	3	-	-	-	-	-	-	-
M23MCAL107.2	-	3	-	-	-	-	-	-
M23MCAL107.3	-	-	3	-	-	-	-	-
M23MCAL107.4	-	-	3	2	-	-	-	-
M23MCAL107	3	-	3	2	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Lab Programs	10	-	-	-	**10**
	-	10	-	-	**10**
	-	-	15	-	**15**
	-	-	-	15	**15**
Total	**10**	**10**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	**CO2**	**CO3**	**CO4**	**Total**
Lab Programs	20	-	-	-	**20**
	-	20	-	-	**20**
	-	-	30	-	**30**
	-	-	-	30	**30**
Total	**20**	**20**	**30**	**30**	**100**

10. Future with this Subject

The future of data communications and networking will be characterized by continuous innovation and adaptation to meet the evolving demands of an increasingly connected world.

- **Cybersecurity**: Increased focus on network security, covering topics such as encryption, intrusion detection, and secure communication.
- **Cloud Computing**: Integration of cloud networking concepts, including virtual networks and cloud service models.
- **Real-World Applications**: Use of practical, real-world scenarios to illustrate network design and troubleshooting.
- **Hands-On Learning**: More interactive and hands-on labs using simulation tools and real networking equipment.

1st Semester	MANDATORY CREDIT COURSE (MC) PROFESSIONAL COMMUNICATON AND SKILL ENHANCEMENT -1	M23MCA108

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic Language Proficiency**	A foundational understanding of the language used for communication (e.g., English proficiency for English courses)
2.	**Reading and Writing Skills**	Ability to read and comprehend texts, and write clearly and coherently.
3.	**Listening and Speaking Skills**	Capacity to understand spoken language and express thoughts and ideas verbally.
4.	**Critical Thinking**	Ability to analyze information, make reasoned judgments, and solve problems effectively.

2. Competencies

S/L	Competency	KSA Description
1.	**Presentation Skills**	• **Knowledge:** Planning and Structuring Presentation • **Skills:** Effective Use of Visual Aids, Overcoming Stage fear • **Attitudes:** Effective Usage of presentation **t**echniques and strategies
2.	**Email and Virtual Communication**	• **Knowledge:** Email, Resume Writing, Online Communication • **Skills:** Letter Writing, Virtual Communication • **Attitudes:** Expressing idea, Flawless Communication
3.	**Professional Communication**	• **Knowledge:** Importance, Basics, purpose & audience, cross cultural communication, Language as a tool • **Skills:** Controlling nervousness & stage Fright, Visual aids in presentation • **Attitudes:** Classification of barriers, Effective Presentation Strategies
4.	**Basic English Vocabulary**	• **Knowledge:** Importance, objectives, characteristics, Vocabulary • **Skills:** Grammar, Parts of Speech, Communication Barriers • **Attitudes:** Perform in a team to make an effective oral / written presentation
5.	**Aptitude**	• **Knowledge:** Number System, Problem Solving, Simple Accounts • **Skills:** Problem solving, Accounts, Logical Skills • **Attitudes:** Easy ways of solving problems, logical thinking

3. Syllabus

PROFESSIONAL COMMUNICATON AND SKILL ENHANCEMENT -1
SEMESTER – I

Course Code	**M23MCA108**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:0:0:2)**	SEE Marks	**50**
Total Number of Lecture Hours	**20 hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**01**

Course Objectives:

- Learn and inculcate concepts of Professional Communication and Ethics
- Skill enhancement of logical and reasoning aspects

Module -1

Presentation Skills: Planning and Structuring a Presentation**,** Effective Use of Visual Aids Engaging the Audience, Techniques and Strategies Overcoming Stage Fear**,** Evaluating Presentation Success, JAM Sessions	L1

Module -2

Assertiveness: Understanding the Difference: Assertiveness vs Aggressiveness, Benefits of Being Assertive Techniques for Assertive Communication, Saying No Politely and Firmly Assertiveness Role-Plays **Email and Virtual Communication** Email Etiquette: Do's and Don'ts Crafting Effective Emails: Clarity, Brevity, and Tone Best Practices for Virtual Meetings (Zoom, Teams,etc.) Virtual Communication Tools Navigating Time Zones, Cultural Differences, and Other Challenges	L1
Module -3	
Professional Communication at Workplace: Group Discussions – Importance, Characteristics, Strategies of a Group \Discussions. Group Discussions is a Tool for Selection. Employment/ Job Interviews - Importance, Characteristics, Strategies of a Employment/ Job Interviews. Intra and Interpersonal Communication Skills - Importance, Characteristics, Strategies of a Intra and Interpersonal Communication Skills. Non Verbal Communication Skills (Body Language) and its importance in GD and PI/JI/EI. Presentation skills and Formal Presentations by Students - Importance, Characteristics, Strategies of Presentation Skills. Dialogues in Various Situations (Activity based Practical Sessions in class by Students Team Work and Collaboration Characteristics of Effective Teams Roles and Responsibilities within Teams Strategies for Collaborative Work Handling Team Conflicts Celebrating Team Successes	L1
Module -4	
Basic English: Communicative Grammar and Vocabulary PART-I: Grammar: Basic English Grammar and Parts of Speech, Articles and Preposition. Question Tags, One Word Substitutes, Strong and Weak forms of words, Introduction to Vocabulary, All Types of Vocabulary– Exercises on it. **Introduction to Communicative English**: Communicative English, Fundamentals of Communicative English, Process of Communication, Barriers to Effective Communicative English, Different styles and levels in Communicative English. Interpersonal and Intrapersonal Communication Skills.	L1
Module -5	
Aptitude: Number System, Divisibility & Remainder, Multiples & Factors, Integers, LCM & HCF, Complete a number Series, Find the Missing Term and Wrong Term. **Simplification:** BODMAS Rule, Approximation, Decimals, Fractions, Surds & Indices **Percentage:** Calculation-oriented basic percentage, Profit and Loss, Successive Selling type, Discount & MP, Dishonest Dealings, Partnerships **Interest:** Simple Interest, Compound Interest, Mixed Interest, Installments. **Data Interpretation**: Approach to interpretation - simple arithmetic, rules for comparing fractions, Calculating (approximation) fractions, short cut ways to find the percentages, Classification of data– Tables, Bar graph, line graph, Cumulative bar graph, Pie graph, Combination of graphs, Combination of table and graphs	L1,L2

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to presentation skills**	• Effective Usage of presentation techniques and strategies • Planning and Structuring Presentation • Effective Use of Visual Aids, Overcoming Stage fear
2	**Week 3-4: Implementing communication skills**	• Expressing idea, Flawless Communication • Email, Resume Writing, Online Communication • Letter Writing, Virtual Communication
3	**week 5 -6 Building confidence in communication**	• Classification of barriers, Effective Presentation Strategies • Importance, Basics, purpose & audience, cross cultural communication, Language as a tool • Controlling nervousness & stage Fright, Visual aids in presentation

4	**Week 5-6: Introduction to writing skills**	• Perform in a team to make an effective oral / written presentation • Importance, objectives, characteristics, Vocabulary • Grammar, Parts of Speech, Communication Barriers
5	**Week 7-8: Developing problem solving and logical reasoning**	• Easy ways of solving problems, logical thinking • Number System, Problem Solving, Simple Accounts • Problem solving, Accounts, Logical Skills

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Scheme of Continuous Internal Examination (CIE): Evaluation of CIE will be carried out in TWO Phases.

Phase	Activity
I	CIE1 is conducted for 30 marks is consolidated to 20 Marks.
II	CIE1 is conducted for 30 marks is consolidated to 20 Marks.
III	CIE1 (20 marks) + CIE2 (20marks) + Attendance (10 marks) = 50 marks 10 marks for attendance will be considered only if students have more than 85% attendance
IV	SIE is conducted for 50 marks (Students are allowed to write SIE provided they have minimum of 50% CIE marks and more than 85% attendance

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Presentation**	Develop their potential and become confident in presentation , usage of visual aids
2	**Professional Communication**	Apply and enhance communication, leadership and interpersonal working skills with professionals
3	**Aptitude/ Logical understanding**	Understand and solve problems covering Quantitative, verbal Ability and Logical Reasoning

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA108.1	Students will acquire basic knowledge of English and develop presentation and interaction skills and also problem analyzing skills

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA108.1	**3**	**-**	**2**	**-**	**2**	**-**	**-**	**3**
M23MCA108	**3**	**-**	**2**	**-**	**2**	**-**	**-**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	**Total**
Module 1	**6**	**6**
Module 2	**6**	**6**
Module 3	**6**	**6**
Module 4	**6**	**6**
Module 5	**6**	**6**
Total	**30**	**30**

Semester End Examination (SEE)

	CO1	**Total**
Module 1	**10**	**10**
Module 2	**10**	**10**
Module 3	**10**	**10**
Module 4	**10**	**10**
Module 5	**10**	**10**
Total	**50**	**50**

10. Future with this Subject

- **Digital Communication**: Greater emphasis on digital tools and platforms, including social media, email, and video conferencing.
- **Cross-Cultural Communication**: Increased focus on understanding and navigating communication across different cultures and global contexts.
- **Soft Skills**: Development of essential soft skills like empathy, active listening, and emotional intelligence.
- **Remote Work Skills**: Training on effective communication in remote and hybrid work environments.
- **Data-Driven Communication**: Use of data analytics to improve communication strategies and understand audience engagement.

1st Semester	**BASIC CREDIT COURSE (BC)** **BASICS OF PROGRAMMING AND COMPUTER ORGANIZATION**	**M23MCA109**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basics of Programming & Computer Organization**	• Bridge course is a non-credit course introduced to the students who are admitted into MCA program from non-computer science background. Students have to secure eligibility by scoring 50% marks in CIE (No SEE for this course). • Exemption for BCA/BSc (computer science) students.

2. Competencies

S/L	Competency	KSA Description
1	**C Programming Basics**	**Knowledge:** • Understanding of C Programming • Knowledge of Data Types, Decision making Statements, Arrays **Skills:** • Ability to apply Data Types, Decision making Statements. • Proficiency in utilizing Control Statements and Arrays. **Attitudes:** • Appreciation for the importance of programming aspects
2	**Structures**	**Knowledge:** • Understanding of structure, declaring structure variables **Skills:** • Structure initialization, operations, array of structures. • Functions, Unions, size of structures **Attitudes:** • Appreciation for usage of structures
3	**Pointers**	**Knowledge:** • Understanding Pointers in C **Skills:** • Functions, Call by Value/ Reference • Recursion, pointers to functions **Attitudes:** • Valuing the importance of pointers in C
4	**Binary System and Combinational Logic**	**Knowledge:** • Understanding the Binary Number System and Conversions • Knowledge of Binary Logic, Digital Logic Gates **Skills:** • Usage of Numbers Conversion, Binary code, storage, registers **Attitudes:** • Learning and understanding basics of digital electronics part
5	**Computer Organization**	**Knowledge:** • Knowledge of basic structure of computer hardware and functional units. **Skills:** • Understanding the performance and peripheral operations of the CPU **Attitudes:** • Learning and understanding the basic structure and peripherals of a computer hardware

3. Syllabus

BASICS OF PROGRAMMING AND COMPUTER ORGANIZATION
SEMESTER – I

Course Code	**M23MCA109**	CIE Marks	**100**
Number of Lecture Hours/Week(L: T: P: S)	**(2:0:0:2)**	SEE Marks	**--**
Total Number of Lecture Hours	**20 hours**	Total Marks	**100**
Credits	**--**	Exam Hours	**--**

Course objectives: This course will enable students to:

- To understand the structure, function, and characteristics of computer systems.
- To understand the design of the various functional units and components of computers.
- To identify the elements of modern instruction sets and their impact on processor design.
- To explain the function of each element of a memory hierarchy

Module -1	
C Programming: decision making, control structures and arrays C Structure, Data Types, Input-Output Statements, Decision making with if statement, simple if statement, the if.-else statement, nesting of if.-else statements, the else.if ladder, the switch statement, the ?: operator, the goto statement, the break statement, programming examples. The while statement, the do...while statement, for statement, nested loops, jumps in loops, the continue statement, programming examples. One dimensional and two-dimensional arrays, declaration and initialization of arrays, reading, writing and manipulation of above types of arrays.	L1
Module -2	
Structures: Defining a structure, declaring structure variables, accessing structure members, structure initialization, copying and comparing structure variables, operations on individual members, array of structures, structures within structures, structures and functions, Unions, size of structures.	L2
Module -3	
Pointers: Pointers in C, Declaring and accessing pointers in C, Pointer arithmetic, Functions, Call by value, Call by reference, Pointer as function arguments, recursion, passing arrays to functions, passing strings to functions, Functions returning pointers, Pointers to functions, Programming Examples.	L2
Module -4	
Binary Systems and Combinational Logic: Digital Computers and Digital Systems, Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, subtraction using r's and r-1 complements, Binary Code, Binary Storage and Registers, Binary Logic, Integrated Circuits, Digital Logic Gates.	L3
Module -5	
Basic Structure of Computer Hardware and Software: Computer Types, Functional Units, Basic Operational Concepts, Bus structure, Software, Performance, Multiprocessing and Multi computers, Machine Instruction: Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Interrupts.	L4

Textbooks:

1. Programming in ANSI C, Balaguruswamy, 7th Edition, McGraw Hill Education
2. C: The Complete Reference, Herbert Schild,4th Edition, McGraw Hill Education
3. Let us C, Yashwant Kanetkar, BPB Publications
4. M.Morris Mano, "Digital Logic and Computer Design", Pearson, 2012.
5. Carl Hamacher, Zvonko Vranesic Safwat Zaky," Computer Organization", 5th edition, Tata McGraw-Hill, 2011

Skill Development Activities Suggested

The students with the help of the course teacher can take up technical activities which will enhance their skill, or the students can interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to C Programming**	• Basic C Programming • Data Types, Decision Making/Control Statements • Implementing basic programs
2	**Week 3-4: Arrays**	• Arrays – different types of arrays • Arrays initialization, declaration, and usage • Implementing various types of arrays
3	**Week 5-6: Structures**	• Structures, Unions • Structure declaration, accessing, size of structures • Implementation of structures and unions
4	**Week 7-8: Pointers**	• Pointers • Pointer declaration and accessing • Applying Pointers call by value/reference
5	**Week 9-10: Binary System and Combinational Logic**	• Binary System and Combinational Logic • Number System conversions, Number Complement, storage and registers, logic gates • Solving problems of number system and logic gates
6	**Week 11-12: Basics of Computer Hardware**	• Basic Structure of computer hardware and software • CPU working principles and software performance • Understanding peripheral structure and machine instructions

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation /Simulation**	Incorporate visual aids like videos/animations/simulation to enhance understanding of basic concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment details

CIE Split up for Professional Course (PC)

Components	Number	Weightage	Max. Marks	Min. Marks
(i) Internal Assessment-Tests (A)	2	50%	25	12.5

(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

NOTE: This course does not contain any credits.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Structure**	To understand the structure, function, and characteristics of computer systems.
2	**Design**	To understand the design of the various functional units and components of computers.
3	**Elements**	To identify the elements of modern instructions sets and their impact on processor design.
4	**Functions**	To explain the function of each element of a memory hierarchy

8. Course Outcomes and Mapping with POs

Sl. No.	Description
M23MCA109.1	Understand the program's flow with the help of control statements and the sequence of code execution and its influence on the overall operation.
M23MCA109.2	Apply programming concepts to develop simple programs to solve specific problems.
M23MCA109.3	Analyse the program execution while also assessing trade-offs among memory storage and retrieval methods.
M23MCA109.4	Evaluate the performance and suitability of various memory hierarchy setups to determine their efficiency and effectiveness.

Mapping of COs with POs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA109.1	3	-	-	-	-	-	-	-
M23MCA109.2	3	-	-	-	-	-	-	-
M23MCA109.3	-	3	-	-	-	-	-	-
M23MCA109.4	-	-	3	-	-	-	-	-
M23MCA109	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

			IA		
	CO1	**CO2**	**CO3**	**CO4**	**Total**
Module 1	5	5	5	5	**20**
Module 2	5	5	5	5	**20**
Module 3	5	5	5	5	**20**
Module 4	5	5	5	5	**20**
Module 5	5	5	5	5	**20**
Total	**25**	**25**	**25**	**25**	**100**

10. **Future with this Subject**

The "Basics of Programming and Computer Organization" course lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of computer science and engineering. Here are some notable contributions:

- **Algorithm Design and Analysis**

  The knowledge gained in this course about data types, control statements, and basic programming constructs is crucial for understanding algorithm design and analysis. Students learn to implement and analyze algorithms, focusing on efficiency and optimization. Mastery of arrays, pointers, and structures enables students to tackle complex algorithmic problems, contributing to a deeper understanding of computational theory and practical problem-solving.

- **Data Structures**

  Understanding the basics of arrays, pointers, and dynamic memory management forms the core foundation for more advanced data structures. Students will be able to implement linked lists, trees, graphs, and hash tables efficiently. This subject prepares students for the rigorous study of data structures, essential for optimizing storage and retrieval operations, which is fundamental in various applications such as databases and information retrieval systems.

- **Operating Systems**

  The concepts of pointers, memory allocation, and structures are directly applicable to understanding how operating systems manage hardware resources. Students will explore process scheduling, memory management, and file systems.

- **Computer Networks**

  The basic programming skills and understanding of data structures gained in this course are vital for studying computer networks. Students will learn to implement networking protocols and understand data transmission techniques.

2nd Semester	PROFESSIONAL CORE COURSE (PC) RELATIONAL DATABASE MANAGEMENT SYSTEM	M23MCA201

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Computer Literacy**	A solid understanding of how computers work, file management, and using software applications is essential.
2	**Fundamentals of Data and Information**	Familiarize yourself with the concepts of data, information, and knowledge. Understand the differences between structured and unstructured data.
3	**Basic Programming Concepts**	While not mandatory, a familiarity with programming concepts can be helpful, especially if you intend to work with databases in a software development context.
4	**Operating System Concepts**	Familiarity with concepts like file systems, memory management, and process scheduling can help you understand how a DBMS interacts with the underlying operating system.
5	**Problem-Solving Skills**	Develop your analytical and problem-solving skills, as designing efficient and effective databases often requires making trade-offs and optimizing for different scenarios.

2. Competencies

S/L	Competency	KSA Description
1	**Data Modeling**	**Knowledge:** Understand the principles of data modeling. **Skills:** Entity-Relationship diagrams (ERDs), **Attitudes:** These concepts help design efficient and organized database.
2	**Relational Algebra and Set Theory**	**Knowledge:** Gain basic knowledge of relational algebra and set theory. **Skills:** The knowledge used to interact with relational databases. **Attitudes:** The foundation of relational databases.
3	**SQL (Structured Query Language):**	**Knowledge:** the basics of SQL, the standard language for data query. **Skills:** Writing queries to retrieve, update, and manipulate data. **Attitudes:** Acquired skill to be used for querying with relational databases.
4	**Normalization**	**Knowledge:** Learn about database normalization. **Skills:** To eliminate redundancy and improve data integrity. **Attitudes:** Understand the concept of normalization for optimizing query performance.
5	**Data Base applications**	**Knowledge:** Gain insight into query optimization strategies. **Skills:** To design data base structure for a particular application. **Attitudes:** To enhance database performance.

3. Syllabus

RELATIONAL DATA BASE MANAGEMENT SYSTEM SEMESTER – II			
Course Code	**M23MCA201**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(4:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**50 hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**
Course objectives: • To provide a strong foundation in database concepts, technology, and practice. • To practice SQL programming through a variety of database problems. • To understand the relational database design principles. • To demonstrate the use of concurrency and transactions in database.			

• To design and build database application for real world problems. • To become familiar with database storage structures and access techniques.	
Module -1	
Introduction to DBMS and Database Design Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications. Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces. The Database System environment. Conceptual Data Modeling using Entities and Relationships: Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization.	L1, L2, L3
Module -2	
Relational Models Relational Model:Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. Relational Algebra and Calculus:Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra, Tuple relational calculus, Domain relational calculus. Mapping Conceptual Design into a Logical Design:Relational Database Design using ER-to-Relational mapping.	L1, L2, L3
Module -3	
SQL SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL : Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.	L1, L2, L3
Module -4	
Normalization: Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	L1, L2, L3
Module -5	
Database Application Development: Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures. **Case study:** The internet Bookshop. Internet applications: The three tier application architecture.	L1, L2, L3
Text Books: 1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson. **2.** Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw-Hill, 3rdEdition. **Reference books:** 1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan‟s Database System Concepts 6th edition Tata McGraw-Hill	

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3	• Understand the principles of data modeling

	Introduction to database and design	• Entity-Relationship diagrams (ERDs). These concepts help design efficient and organized database.
2	Week 4-6 **Relational Database**	• Gain basic knowledge of relational algebra and set theory. • The knowledge used to interact with relational databases and the foundation of relational databases.
3	Week 7-9 **SQL**	• The basics of SQL, the standard language for data query. • Writing queries to retrieve, update, and manipulate data.
4	Week 10-12 **Normalization**	• Learn about database normalization to eliminate redundancy and improve data integrity. • Understand the concept of normalization for optimizing query performance.
5	Week 13-15 **Database Application Development**	• Gain sight into query optimization strategies to enhance database performance. • To design data base structure for a particular application.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Using traditional lecture methods and ICT as and when needed.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance learning.
3	**Collaborative Learning**	Encourage collaborative learning approaches for peer learning.
4	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application.
5	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
6	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Course (PC)

	Components	Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Introduction to database and design**	To provide a strong foundation in database concepts, technology, and practice.

2	**SQL**	To practice SQL programming through a variety of database problems.
3	**RDBMS**	To understand the relational database design principles.
4	**Database Application Development**	To design and build database application for real world problems.
5	**Database Storage**	To become familiar with database storage structures and access techniques.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs):

COs	Description
M23MCA201.1	Understand and apply the basic elements of a relational database management system.
M23MCA201.2	Apply various constraints, techniques and Structured Query Language (SQL) statement for database operations.
M23MCA201.3	Analyze various database models and normalization for the given application.
M23MCA201.4	Design and develop entity relationship model and database application using modern tools

CO-PO Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA201.1	3	-	-	-	-	-	-	-
M23MCA201.2	3	-	-	-	-	-	-	-
M23MCA201.3	-	3	-	-	-	-	-	-
M23MCA201.4	-	-	3	3	-	-	-	-
M23MCA201	3	3	3	3	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**5**				**5**
Module 2	**5**	**5**			**10**
Module 3		**5**	**5**		**10**
Module 4			**10**	**5**	**15**
Module 5				**10**	**10**
Total	**10**	**10**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**10**	**10**			**20**
Module 3		**10**	**10**		**20**
Module 4			**20**	**10**	**30**
Module 5				**20**	**20**
Total	**20**	**20**	**30**	**30**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject:

- Data Organization and Storage: Companies can store their data in databases in a structured, organized manner, making it simpler to access and analyze.
- Data Analysis: Databases contain a lot of data, and with the correct tools, organizations can analyze that data to find insights that will help them make business decisions and strategies.
- Efficiency: Databases give companies a centralized area to keep their data, making it more straightforward for staff to retrieve the data they want, minimizing duplication of work and boosting efficiency.
- Security & Privacy: Databases let companies control who has access to their data, ensuring that only authorized users may see and change it. This aids in preventing unauthorized access to and breaches of vital consumer and corporate information.
- This course is the foundation for many other courses to follow such as cloud storage, distributed data storage, block chain, Big data, Quantum computing etc.,

2nd Semester	**Professional Core Course (PC)** **OBJECT ORIENTED PROGRAMMING USING JAVA**	**M23MCA202**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Computer Skills**	Basic computer skills, such as saving files in multiple versions and formats.
2	**Programming Fundamentals**	Familiar with general coding concepts like variables, basic data types, Conditional Statements, Looping, Functions, creation of source file, compilation process, program execution techniques.
3	**Multi-Process Execution Programming**	Familiar with the way to execute multiple tasks simultaneously by creating multiple processes.
4	**Basic Object Orientation Concepts**	Basic of four basic principles: encapsulation, inheritance, polymorphism, and abstraction. Where these four OOP principles can be used enable to create objects and collaborate to create powerful applications too.
5	**Programming basic tools**	Familiar with Programming tools like assemblers, compilers, linkers translate, flowchart, algorithms which can be used to form a program from a human write-able and readable source language into the bits and bytes that can be executed by a computer.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Object Oriented Concepts**	**Knowledge:** • Importance of Object Orientation Concepts. • Understanding of the basics of Object Orientation Programming. **Skills:** • Ability to apply Object Orientation Concepts to create objects using appropriate structure. **Attitudes:** • Appreciation to understand the importance of object orientation perspective and implement the same at basic level.
2	**Basic of Programming**	**Knowledge:** • Understanding of basic elements of programming specific to Java Language. • Basics of Java program execution. **Skills:** • Designing basic Java program using basic elements of programming language. • Creating and executing simple Java programs. **Attitudes:** • Appreciation for the role of Java programming elements and its execution.
3	**Java Classes and its methods**	**Knowledge:** • Understanding how classes are defined with data members and methods. **Skills:** • Designing of classes for real world objects. • Defining appropriate attributes and methods for classes. **Attitudes:** • Valuing the importance of classes and its methods in line with real-world objects.
4	**Reusability of**	**Knowledge:**

	Classes and Methods	• Understanding the importance of code reusability through classes and methods reusability. **Skills:** • Applying concepts of object orientation with classes and methods. • Describing the actually importance of reusability through implementations. **Attitudes:** • Openness to learning and using object orientation concepts to achieve code reusability.
5	**Exceptions and Handling the Exceptions**	**Knowledge:** • Understanding of issues with exceptions. **Skills:** • Implementing how to handle the exceptions through appropriate Java programming construct. **Attitudes:** • Appreciation for the way exception is handled and making the execution of program in control.
6	**Multi-Threaded Programming**	**Knowledge:** • Understanding the characteristics and importance of parallel execution of a task. **Skills:** • Designing and analyzing the parallel execution using thread concepts. • Implementing multi-thread concepts. **Attitudes:** • Recognizing the significance of flip-flops in sequential logic circuits

3. Syllabus

OBJECT ORIENTED PROGRAMMING USING JAVA SEMESTER – II			
Course Code	**M23MCA202**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(4:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**50 hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**
Course objectives: This course will enable students to: • To learn primitive constructs JAVA programming language. • To understand Object Oriented Programming Features of JAVA. • To gain knowledge on: packages, multithreaded programing and exceptions. • Create applications using advanced features of JDBC and implement projects .			
Module -1			
Introduction to Java: What is java, Goals of Java Technology, *Similarities and Difference between C++ and Java,* JVM , Garbage Collection , JRE,JIT, Java Debugger, Class loader , Byte code verification , Simple application on java, Compile time Error and run time Error, IDEs, Auto-boxing **Object Oriented Programming:** Software Engineering, The Analysis and design phase Abstraction, class as a blue print. Declaring class, variables and Methods Accessing object Members, Information Hiding . Encapsulation, Declare Constructors and default Constructors. Source file Layout , packages, Compile using – d option Design Tools (Argo UML)			L1
Module -2			
Identifiers, Key Words and Types: Semicolon, block and whitespace. Identifiers, Keywords, Data Types Java Reference Type, Constructing and initializing object. Java Reference type , Memory allocation , This keyword and pass by value . Assigning variable and reference **Expression and Flow Control**: Variable scope, operators, Bitwise operators , Right shift and left shift operators String concatenation, casting. Conditional Statements and loops in java			L1

Arrays: Declaring, creating, Initializing Arrays. Multidimensional Arrays Array Bounds Enhance for loop, Array resizing and Array copy	
Module -3	
Class Design: Sub classing, single inheritance, Access control Overriding methods, Polymorphism, Polymorphic Objects Instance of keyword, casting objects, overloading functions, Variable arguments methods Overloading constructors, Invoking parent class constructors ,Object class, Equals method, Wrapper class , Types of executable methods(jar,exe) Creation of Junit Classes **Advance Class Design:** Static and final keyword Static initializes , final variables, Enumeration type, static import ,and Abstract classes, Interface **Exception Handling:** Exception and Assertions Try catch and finally block Exception categories Method Overriding and Exceptions Creating and Handling User defined Exceptions Assertions.	L2
Module -4	
Collection and Generics Framework: Collection API, List, set, Map. Comparable and comparator Interface Array list, linked list Generics . Enhance for loop. **IO Fundamentals and Files Operations:** Command line arguments , System properties IO Fundamentals , Input Stream and Output Stream Reader and Writer Class. Files operations and its classes Serialize Date class and De- serialize Date class **Building GUI and Event handling:** GUI using AWT and JFC SWING Package. Creating menu bar , menu and Menu Items . Event handling Techniques Evening handling using Anonymous classes and inner classes.	L3
Module -5	
Threads: What is thread, creating and starting a thread. Life cycle of a thread .Thread scheduling Termination a thread and basic controls on thread. Synchronized Keyword. Object lock flag and relies lock flag Deadlock stage, Wait and notify method , Join and yield methods, **Networking:** Networking basics, Socket class and server socket class .Client program and server program. **RDBMS**: Introduction to Relational database management System Query and Statements CRUD operation with any database. **JDBC API:** Introduction to Relational database management System Query and Statements CRUD operation with any database.	L3

Text Books:

1. JAVA: The Complete Reference, 8thEdition, by Herbert Schildt, November 2012, McGraw-Hill Edition 2011, ISBN:978-1-25-900246-5.

Reference Books:

1. Programming with Java A Primer, 4th Edition, by E Balagurusamy, Mar-2010, Tata McGraw Hill Education, ISBN:978-0-07-014169-8.
2. Programming with JAVA, 5th Edition, by M P Bhave and S A Patekar, 2017, Pushp Print Services, ISBN:978-81-317-2080-6.

Tutorial Components

1. Write and Execute a Java program to show how the different ways of declaring and initialization a Two-Dimensional array in Java.
2. Write and Execute a Java program to print list of student names using for-each loop.
3. Develop a class called Student with the data members USN, Name, IA1_Marks, IA2_Marks, IA_2 Marks and Avg_Marks and method ComputeAvg(m1,m2,m3) to compute the average of IA Marks. Develop the suitable class and main method for demonstration.
4. Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager, Developer, and Programmer. Each subclass should have properties such as name, address, salary, and job title. Implement methods for displaying

salary with 10% raise for Programmer, 25% raise for Developer and 40% raise for Manager.

5. Write and Execute a Java program to show the order of constructor call and its execution in multi-level inheritance.
6. Write a Java program to create an interface Sortable with a method Sort() that sorts an array of integers in ascending order. Create two classes BubbleSort and SelectionSort that implement the Sortable interface and provide their own implementations of the Sort() method.
7. Demonstrate how MyPack package is created in Java with class called MyClass and method called MyMethod() and import the package MyPack in the file called New.java to declare object for the class MyClass and call the method MyMethod() in the main method of New.java file.
8. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.
9. Create a child thread by implementing the Runnable interface wherein the child thread does string concatenation, and the main thread changes the string to uppercase.
10. Write a Java program to Create three classes Storage, Counter and Printer. The Storage class should store an integer, the Counter class should create a thread that starts counting from 0 (like, 0,1,2,3,…) and stores each value in the Storage class. The Printer class should create thread that keeps reading the value from the Storage class and prints it.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to Java, OOP**	Basic Java Programming, Java Programming basic constructs and applying basic programming constructs in Java execution environment. Object Oriented Programming Concepts
2	**Week 3-4: Identifiers, Keywords, Types, Expression and Control flow, Arrays**	Java identifiers, keywords data types, java reference type, operators, loops in java, arrays, multi-dimensional arrays.
3	**Week 5-6: Class Design, Advance Class Design, Exception Handling**	Class Methods with Polymorphism and Access Control, using methods in Java Classes and accessing the members and class using appropriate access control with polymorphism and designing and implementing class methods through polymorphism and access mechanism. Exception and Assertions, Try Catch and finally block
4	**Week 7-8: Collection and Generics framework, IO Fundamentals and File Operations**	Collection API Array list Generics, IO Fundamentals File Operations and Serialize Date Class
5	**Week 9-10: Building GUI Event handling**	GUI AWT and JFC SWING Package, Event Handling Techniques Event Handling using Anonymous classes and inter classes
6	**Week 11-12: Threads, Networking, RDBMS, JDBC API**	Understanding multi-threaded concepts with synchronization and inter-thread communications and networking basics, socket class, RDBMS-CRUD operation with any database and JDBC API

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Image/Video/Animation**	Incorporate visual aids like image/videos/animations to enhance understanding of programming constructs.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Programming-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Course (PC)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2(TW0) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding basic Java Programming**	Students will grasp the fundamental concepts of Java Programming, including basic constructs.
2	**Designing simple basic Programs**	Students will learn to design and implement basic and simple Java programs.
3	**Proficiency in Java**	Students will become proficient in understanding and applying the Java specific constructs to improve the efficiency of Java programming logics.
4	**Programming-Based Learning**	Through program execution-based learning, students will undergo the demonstration of Java programming constructs working principles.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA202.1	Understand and apply the basic programming constructs.
M23MCA202.2	Apply the structure of classes and methods in Java programming environment.
M23MCA202.3	Analyze the different programming constructs of Java and its effectiveness in improving the efficiency of Java programs.
M23MCA202.4	Implement appropriate Java programming constructs to solve real-world problem sample scenarios.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA202.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA202.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA202.3	**-**	**3**		**-**	**-**	**-**	**-**	**-**
M23MCA202.4	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA202	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**5**	**10**			**15**
Module 3		**5**			**5**
Module 4			**10**		**10**
Module 5				**10**	**10**
Total	**15**	**15**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2	**10**	**20**			**30**
Module 3		**10**			**10**
Module 4			**20**		**20**
Module 5				**20**	**20**
Total	**30**	**30**	**20**	**20**	**100**

Conditions for SEE Paper Setting:
Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

Continued Popularity: Java remains one of the most popular programming languages, particularly in enterprise environments. Its strong support for OOP will ensure it remains relevant.

Modern Features: Java continues to evolve with each new version, adding features that make OOP more efficient and powerful. For instance, recent versions have introduced enhancements like records, sealed classes, and pattern matching.

Integration with Functional Programming: Java is increasingly incorporating functional programming features, such as lambdas and the Stream API, allowing for a blend of OOP and functional programming paradigms.

Micro services and Cloud Computing: The rise of micro services architecture and cloud computing has led to a shift in how Java applications are developed and deployed. Java's robust ecosystem supports these trends, ensuring that OOP principles can be effectively applied in modern, distributed systems.

Performance Improvements: Ongoing performance improvements in the Java Virtual Machine (JVM) and the language itself will continue to make Java a strong choice for high-performance applications.

Community and Ecosystem: Java benefits from a large and active community, as well as a rich ecosystem of libraries and frameworks that support OOP. This community-driven development will continue to enhance Java's capabilities and ensure its relevance.

2nd Semester	**Professional Core Course (PC)** **AGILE SOFTWARE ENGINEERING**	**M23MCA203**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Understanding Agile Principles**	Familiarity with the Agile Manifesto and its core principles. Knowledge of common Agile frameworks (Scrum, Kanban, XP).
2.	**Team Mindset and Culture**	Commitment to collaborative and cross-functional teamwork. Willingness to embrace change and continuous improvement. Culture of trust and open communication.
3.	**Stakeholder Buy-in**	Support from management and stakeholders for Agile practices. Clear understanding of Agile benefits and how they align with business goals.
4.	**Training and Education**	Training for all team members on Agile methodologies and practices. Ongoing education and coaching to reinforce Agile concepts.
5.	**Agile Roles and Responsibilities**	Clearly defined roles such as Product Owner, Scrum Master, and Development Team. Understanding of each role's responsibilities and interdependencies.
6.	**Effective Communication Tools**	Use of collaboration tools like Jira, Trello, Confluence, Slack, or others. Tools for continuous integration/continuous deployment (CI/CD).

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Agile Software Development**	**Knowledge** • **Agile Principles and Manifesto**: Understanding the core values and principles outlined in the Agile Manifesto with history and evolution of Agile methodologies. • **Agile Frameworks**: Knowledge of different Agile frameworks such as Scrum, Kanban, Lean, and Extreme Programming (XP). • **Software Development Life Cycle (SDLC)**:Basic understanding of the stages in the SDLC and how Agile differs from traditional (waterfall) approaches. **Skills** • **Communication**: Proficiency in clear and concise verbal and written communication, ensure effective information sharing within the team. • **Time Management**: Skills in managing time effectively to meet iteration goals, deadlines and prioritization of tasks. • **Problem-Solving**: Analytical skills to identify and resolve issues that arise during development. • **Adaptability**: Flexibility to adapt to changing requirements and evolving project landscapes. **Attitudes** • **Learning Agility**: Ability to quickly grasp new concepts and practices related to Agile methodologies. • **Facilitation and Leadership**: Ability to lead meetings and Agile ceremonies effectively.
2	**Time and Measures**	**Knowledge** • **Agile Metrics and KPIs**: Understanding key Agile metrics such as velocity, cycle time, lead time, burn-down charts, and burn-up charts. • **Time-Boxing**: Knowledge of the concept of time-boxing and its importance

		in Agile methodologies. - **Estimation Techniques**: Awareness of different estimation methods like story points, planning poker, T-shirt sizing, and affinity estimation. - **WIP (Work in Progress) Limits**: Understanding the concept of WIP limits in Kanban and how they help manage flow and reduce bottlenecks. **Skills** - **Effective Time Management**: Skills in managing time efficiently to ensure timely completion of sprints and tasks. - **Estimation and Planning**: Ability to accurately estimate the effort required for user stories and tasks. Skills in creating realistic and achievable sprint plans and release plans. - **Tracking and Monitoring**: Proficiency in using tools (e.g., Jira, Trello) to track progress, manage backlogs, and monitor team performance. Ability to analyze and interpret Agile metrics to assess team productivity and identify areas for improvement. **Attitudes** - **Attention to Detail**: Ability to maintain a high level of accuracy in tracking time, progress, and metrics. - **Analytical Thinking**: Ability to analyze metrics and performance data to derive meaningful insights. Skills in identifying patterns and trends that can inform decision-making and process improvements. - **Adaptability and Flexibility**: Ability to adapt plans and schedules based on changing requirements and priorities.
3	**Planning, Trust and Team**	**Knowledge** - **Agile Planning Principles**: Understanding the iterative and incremental nature of Agile planning. Knowledge of short-term (sprint planning) and long-term (release planning) strategies. - **Team Dynamics and Collaboration**: Awareness of group dynamics and stages of team development (forming, storming, norming, performing). Understanding the importance of cross-functional teams and the roles within an Agile team (Product Owner, Scrum Master, Development Team). **Skills** - **Effective Planning**: Proficiency in creating and maintaining a product backlog with well-defined user stories. Skills in conducting sprint planning sessions to ensure achievable and clear sprint goals. - **Prioritization and Estimation**: Skills in prioritizing tasks based on value, dependencies, and effort. Ability to use estimation techniques such as story points, planning poker, and affinity estimation. - **Communication and Transparency**: Strong communication skills to ensure clear and open exchange of information within the team and with stakeholders. - **Conflict Resolution**: Skills in mediating conflicts and facilitating constructive discussions to resolve issues. **Attitudes** - **Empathy and Emotional Intelligence**: Ability to understand and empathize with team members' perspectives and emotions. Skills in using emotional intelligence to navigate and influence team dynamics positively. - **Leadership and Mentorship**: Ability to lead by example and inspire trust and confidence within the team. - **Collaboration and Team Building**: Strong ability to foster a collaborative environment where team members feel valued and heard. Skills in building a sense of community and shared purpose within the team.

		• **Accountability and Ownership**: Ability to take ownership of tasks and encourage team members to do the same. Commitment to accountability and delivering on promises.
4	**Reflection and Leadership**	**Knowledge** • **Agile Reflection Practices**: Understanding the purpose and importance of retrospectives in Agile. Knowledge of various retrospective techniques and formats to facilitate effective reflection. Awareness of the role of continuous improvement in Agile methodologies. • **Leadership Principles in Agile**: Knowledge of servant leadership and its application in Agile environments. Understanding of transformational leadership and its impact on team dynamics and performance. Familiarity with leadership roles within Agile frameworks, such as the Scrum Master and Product Owner. **Skills** • **Facilitation of Retrospectives**: Skills in planning and conducting retrospectives to ensure productive reflection. Ability to use various retrospective formats to engage the team and elicit valuable insights. • **Effective Communication**: Strong communication skills to convey vision, goals, and feedback clearly and constructively. Ability to facilitate open and honest communication within the team. • **Decision-Making**: Ability to make informed decisions based on team input and data. Skills in balancing short-term needs with long-term goals in decision-making processes. **Attitudes** • **Empathy and Emotional Intelligence**: Ability to understand and empathize with team members' perspectives and emotions. Skills in using emotional intelligence to navigate and influence team dynamics positively. • **Inspirational Leadership**: Ability to inspire and motivate team members towards achieving common goals. Capacity to create a shared vision and foster a sense of purpose within the team. • **Commitment to Continuous Improvement**: Strong focus on fostering a culture of continuous improvement within the team. Ability to lead by example in seeking feedback and striving for personal and team growth.

3. Syllabus

<table>
<tr><td colspan="5"><b>AGILE SOFTWARE ENGINEERING
SEMESTER – II</b></td></tr>
<tr><td>Course Code</td><td><b>M23MCA203</b></td><td>CIE Marks</td><td colspan="2"><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week(L: T:P:S)</td><td><b>(3:0:0:0)</b></td><td>SEE Marks</td><td colspan="2"><b>50</b></td></tr>
<tr><td>Total Hours</td><td><b>40 hours</b></td><td>Total Marks</td><td colspan="2"><b>100</b></td></tr>
<tr><td>Credits</td><td><b>03</b></td><td>Exam Hours</td><td colspan="2"><b>03</b></td></tr>
<tr><td colspan="5"><b>Course objectives:</b> This course will enable students to:
● Ability to understand agile development processes and the principles behind the agile development.
● Analyze the different perspective related to time and measure different components in project development.
● Learn how to incorporate quality, learning, abstraction components in the software.
● Understand the importance of team and leadership component software development.</td></tr>
<tr><td colspan="5"><b>Module -1</b></td></tr>
<tr><td colspan="4"><b>Introduction to Agile software Development:</b> Overview, Three Perspective on Software engineering, The Agile Manifesto, Individuals and Interactions over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract,</td><td></td></tr>
</table>

Responding to change over a plan, Application of Agile Software Development. **Teamwork:** Overview, A Role Scheme in Agile Teams, Human Perspective on the Role Scheme, Using the Role Scheme to Scale Agile Projects. **Customers and Users:** Overview, The Customer, Customer Role, The User, Combining UCD with Agile Development.	**L1,L2**
Module -2	
Time and Measures: **Time:** Overview, Time-Related Problems in Software Projects, List of Time-Related Problems of Software Projects. The Time Perspective, Tightness of Software Development Methods, Sustainable Pace, Time Management of Agile Projects. **Time Measurements:** Why Are Measures Needed, Who Decides What Is Measured? What Should Be Measured, When Are Measures Taken? How Are Measures Taken? Who Takes the Measures? How Are Measures Used?	**L1,L2**
Module -3	
Quality, Learning and Abstraction: **Quality:** Overview, The Agile Approach to Quality Assurance, Process Quality, Product Quality, Test-Driven Development. **Learning**: Overview, Agile Software Development from the Constructivist Perspective, The Role of Short Releases and Iterations in Learning Processes, Reflection. **Abstraction** :Overview, Objectives, Study Questions, Abstraction Levels in Agile Software Development, Roles in Agile Teams	**L1,L2**
Module -4	
Planning, Trust and Globalization **Planning** :The Stand Up Meeting, Design and Refactoring, Abstraction in Learning Environments, **Trust**: Overview, Process Transparency, Ethics, Diversity. **Globalization:** Overview, Objectives, The Agile Approach in Global Software Development, Software projects and Culture. Planning in distributed agile projects, tracking agile distributed projects.	**L1,L3**
Module -5	
Reflection , Change and Leadership **Reflection:** Overview, Reflection on Learning in Agile Software Development, Reflective Practitioner Perspective, Retrospective, The Retrospective Facilitator, Guidelines for a Retrospective Session, End of the Release Retrospective. **Change:** Overview, A Conceptual Framework for Change Introduction, Changes In Software Requirements, Organizational Changes, Transition to an Agile Software Development Environment. **Leadership:** Overview, Objectives, Leaders, Leadership Styles, The Agile Change Leader, Coaches, Delivery and Cyclicality: Overview, Objectives, Delivery, Towards the End of the Release, Release Celebration, Reflective Session Between Releases.	**L2,L3**

Text Books:

1. Orit Hazzanand Yael Dubinsky, Agile Software Engineering, Springer,2014
2. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education

Reference Books:

1. Agile software development, a list air cockburn, pearson educationIndia
2. Agile estimating and planning, mike cohn, pearson educationIndia;1stedition,2006
3. Michelesliger, staciabroderick,thesoftwareprojectmanager'sbridgetoagility,addison-wesleyprofessional,2008

Journals/Magazines:

1. https://hbr.org/2016/05/embracing-agile
2. https://www.inderscience.com/jhome.php?jcode=ijasm
3. Agile-thoughts:MonthlyAgileMagazine&CommunityHub.https://www.agile-thoughts.com/.

Web/Digital Resources:

1. www.allaboutagile.com/what-is-agile-10-key-principles/
2. https://www.versionone.com/agile
3. Lecture-26Agile Development:https://www.youtube.com/watch?v=jRs-aFETAXY
4. https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/

4. **Syllabus Timeline**

S/L	Syllabus Timeline	Description
1	Week 1-2: **Introduction to Agile software Development**	Understanding the fundamental concepts of agile software development History and evolution of agile software development Different types of agile software development Understanding the purpose and role of agile methodology in software development.
2	Week 3-4: **Time and Measures**	Objectives, Time-Related Problems in Software Projects, List of Time-Related Problems of Software Projects Time Management of Agile Projects, Time Measurements, Prioritizing Development Tasks Software Projects development and the Time Perspective
3	week 5 -6 **Quality, Learning and Abstraction**	Understanding the importance of agile approach to quality assurance, process quality and product quality The Role of Short Releases and Iterations in Learning Processes, Gradual Learning Process of Agile Software Engineering Learning Process of Agile Software Engineering, Reflection, Abstraction
3	Week 5-6: **Planning, Trust and Team**	Efficiently managing memory resources in an operating system. Memory hierarchy and organization. Virtual memory concepts and techniques. Implement memory allocation strategies such as paging and segmentation. Configure and manage virtual memory systems.
4	Week 7-8: **Reflection and Leadership**	Reflection on Learning in Agile Software Development, Reflective Practitioner Perspective, Retrospective The Retrospective Facilitator, Guidelines for a Retrospective Session, End of the Release Retrospective. Change Overview, Objectives, Delivery, Towards the End of the Release, Release Celebration, Reflective Session Between Releases

5. **Teaching-Learning Process Strategies**

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS)**	Pose HOTS questions to stimulate critical thinking related to each competency.

	Questions	
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details

CIE Split up for Professional Course (PC)

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objective

S/L	**Learning Objectives**	**Description**
1	**Understand Agile Principles**	Learn the core values and principles of Agile, such as flexibility, collaboration, and customer focus.
2	**Work in Iterations**	Understand how to break down projects into small, manageable pieces (iterations) that can be completed in short timeframes.
3	**Collaborate Effectively**	Learn to work closely with team members and stakeholders, ensuring open communication and teamwork.
4	**Adapt to Change**	Develop the ability to quickly respond to changes in requirements, even late in the project.
5	**Deliver Incremental Value**	Focus on delivering functional software regularly, with each iteration providing a usable piece of the final product..
6	**Continuous Improvement**	Embrace a mindset of ongoing learning and improvement, regularly reflecting on processes and making adjustments.
7	**Quality Focus**	Learn techniques for maintaining high quality, such as test-driven development, continuous integration, and regular reviews.

8. Course Outcomes and Mapping with POs

CO's	DESCRIPTION OF THE OUTCOMES
M23MCA203.1	Understand the concept to agile software development
M23MCA203.2	Analyze and apply the time and measures related perspective to agile software development.
M23MCA203.3	Review and design the different agile approaches to quality assurance, Learning Processes and abstraction levels in Agile Software Development.
M23MCA203.4	Identify the different approaches related to planning, trust process Transparency, and team work.

Course Outcomes mapping to Program Outcomes:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA203.1	3	-	-	-	-	-	-	-
M23MCA203.2	2	3	-	-	-	-	-	-
M23MCA203.3	-	-	3	-	-	-	-	-
M23MCA203.4	-	-	-	2	3	-	-	-
M23MCA203	2.5	3	3	2	3	-	-	-

9. Assessment Plan

Continuous Internal Evaluation - CIE

	CO1	CO2	CO3	CO4	Total
Module 1	10				**10**
Module 2	5	5			**10**
Module 3		10			**10**
Module 4			10		**10**
Module 5				10	**10**
Total	**15**	**15**	**10**	**10**	**50**

Semester End Examination -SEE

	CO1	CO2	CO3	CO4	Total
Module 1	20				**20**
Module 2	10	10			**20**
Module 3		20			**20**
Module 4			20		**20**
Module 5				20	**20**
Total	**30**	**30**	**20**	**20**	**100**

10. Future with this Subject

The future of agile software engineering is influenced by several emerging trends and evolving practices that are shaping how software development teams operate and deliver value. Here are some key aspects of the future landscape of Agile software engineering:

- **DevOps Alignment**: Agile methodologies are closely aligned with DevOps practices to streamline the end-to-end software delivery lifecycle. This integration emphasizes collaboration, automation, and continuous delivery of software updates.

- **Continuous Integration/Continuous Deployment (CI/CD)**: Agile teams are embracing CI/CD pipelines to automate build, test, and deployment processes, enabling faster feedback loops and more frequent releases.

- **Enterprise Agile Frameworks**: Scaling Agile beyond individual teams to entire organizations is facilitated by frameworks like SAFe (Scaled Agile Framework), LeSS (Large-Scale Scrum), and Nexus. These frameworks provide guidance on coordinating multiple Agile teams, aligning with business objectives, and managing dependencies.
- **Combining Agile with Traditional Methods**: Hybrid Agile approaches blend Agile methodologies with elements of traditional project management frameworks. This flexibility allows organizations to adapt Agile practices to suit complex project requirements and diverse team structures.
- **Business Agility**: Agile software engineering emphasizes delivering value to customers quickly and iteratively. Future trends will continue to prioritize business agility, enabling organizations to respond swiftly to market changes and customer needs.
- **Value Stream Optimization**: Agile teams are adopting Lean principles to optimize value streams, eliminate waste, and improve the flow of value delivery from concept to deployment.
- **Remote Work**: The rise of remote work has accelerated the adoption of Agile practices in distributed teams. Agile methodologies are evolving to support effective collaboration, communication, and team cohesion in virtual environments.
- **Virtual Agile Practices**: Tools and platforms that facilitate virtual Agile ceremonies, collaborative planning sessions, and real-time communication are becoming essential for remote Agile teams.
- **Servant Leadership**: Agile leaders embrace servant leadership principles, focusing on empowering teams, removing impediments, and fostering a culture of trust, autonomy, and continuous improvement.
- **Empowering Self-Organizing Teams**: Agile software engineering encourages self-organizing teams that have the authority and responsibility to make decisions, adapt to change, and deliver valuable software increments.

2nd Semester	**INTEGRATED PROFESSIONAL CORE COURSE (IPC) PYTHON PROGRAMMING**	**M23MCA204**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Science**	• **Logical Reasoning**: Ability to follow logical processes, identify patterns, and apply basic problem-solving strategies. • **Basic Electronics Awareness:** Understanding of binary data and simple electronic concepts such as voltage and current (not mandatory, but helpful).
2	**Mathematics**	• **Algebra:** Proficiency in solving linear equations and inequalities, which will assist in understanding programming logic**.** • **Set Theory:** Familiarity with basic set operations (union, intersection), which are relevant in data manipulation tasks**.**
3	**Computer Science**	• **Basic Programming Knowledge:** Experience in writing simple code in any programming language (preferably Python), with an understanding of variables, loops, and conditionals. • **Problem-Solving Skills:** Ability to approach and decompose problems methodically.
4	**Data Structures**	• **Data Organization Basics:** Awareness of how data can be stored and retrieved using simple structures like lists or arrays. • **Basic Algorithmic Concepts:** Understanding the importance of sorting, searching, and basic operations on data.
5	**Object-Oriented Programming**	• **Introduction to OOP Concepts:** Basic understanding of classes and objects, and how they represent real-world entities. Prior experience in any OOP language is beneficial but not essential.
6	**Graphical User Interface Development**	• **GUI Fundamentals Awareness:** General knowledge of what a graphical user interface is and its significance in software development. Prior exposure to GUI tools is optional but advantageous.

2. Competencies

S/L	Competency	KSA Description
1	**Basics of Python Programming**	**Knowledge:** Understanding of Python's syntax, data types, variables, operators, expressions, statements, and control flow mechanisms (sequence, selection, iterations). **Skills**: Ability to write, debug, and execute basic Python programs, handle input/output operations, and apply control flow statements effectively. **Attitudes**: Detail-oriented approach to coding, willingness to experiment with code, and persistence in debugging and refining solutions.
2	**Functions in Python**	**Knowledge:** In-depth understanding of functions, including defining, calling, and passing parameters. Familiarity with return values, void functions, recursive functions, and exception handling. **Skills:** Proficiency in writing and using functions, handling different types of function parameters, and implementing recursion and exception handling. **Attitudes:** Focus on modular programming practices, appreciation for reusable code, and thoroughness in testing and debugging functions.
3	**Collection Data Types**	**Knowledge:** Comprehensive understanding of Python's collection data types such as strings, lists, tuples, sets, and dictionaries, including their methods and operations. **Skills:** Ability to perform operations on these data structures, such as indexing,

		slicing, concatenation, sorting, and manipulating elements**.** **Attitudes:** Careful handling of data, methodical approach to data manipulation, and attention to the efficiency of data operations.
4	**Object-Oriented Programming**	**Knowledge:** Deep understanding of object-oriented principles, including classes, objects, inheritance, polymorphism, encapsulation, and abstraction. Knowledge of constructors, methods, and method resolution order (MRO). **Skills:** Ability to design and implement class structures, apply inheritance and polymorphism, and manage class hierarchies effectively**.** **Attitudes:** Commitment to creating maintainable and scalable code, emphasis on clean design and reusable components, and willingness to refactor and improve code as needed.
5	**Graphical User Interface Development**	**Knowledge:** Basic understanding of GUI design principles and the tkinter library for Python, including widgets, windows, frames, and event handling. **Skills:** Ability to create and manage GUI components, handle user interactions, and design user-friendly interfaces. **Attitudes:** Creativity in designing intuitive interfaces, patience in refining user experiences, and enthusiasm for exploring GUI development.
6	**Python Modules**	**Knowledge:** Understanding of Python modules and packages, including how to create, import, and use them effectively**.** Familiarity with specialized libraries in different domains like (NumPy, Pandas, OS, SYS) **Skills:** Proficiency in organizing code into modules, leveraging built-in libraries, and creating custom modules for various functionalities**.** Ability to apply these libraries effectively in projects, integrate them into applications, and leverage their functionalities for practical use cases. **Attitudes:** Openness to utilizing and integrating various Python libraries and tools, and a proactive approach to learning and applying new modules.

3. Syllabus

PYTHON PROGRAMMING

SEMESTER – II

Course Code	**M23MCA204**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:2)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course Objectives:

1. To understand Python syntax, data types, variables, operators, expressions, and control flow statements.
2. To learn how to define, call, and use functions, including handling parameters, return values, and exceptions.
3. To analyze Python's collection data types—strings, lists, tuples, sets, and dictionaries—along with their operations and methods
4. To explore object-oriented programming principles such as classes, objects, inheritance, and polymorphism, and apply them in Python
5. To build interactive graphical user interfaces using the tkinter library, including windows, widgets, and event handling.
6. To learn about creating, importing, and using Python modules and packages to organize and reuse code.
7. To explore and apply various Python libraries for different domains, such as data science, web development, automation, and more.

Module-1

Basics of Python Programming: Introduction, Data types, Identifiers, Keywords, Operators, Variables, Expressions, Statements, Indentations, Type Conversions, Input/ Output operators, Math modules, **Control flow statements:** Sequence, Selection, Iterations in python, Control Flow Modifiers

Module-2

Functions: Introduction to Function, Calling functions, Function parameters, Void functions, Return Values, recursive functions, default parameters, Lambda functions & map**. Exception handling –** Exception handling with try, handling multiple exceptions, writing you own exception. **Strings-** Basics, methods, String Formatting, String Slicing and Indexing

Module-3

Storage structures / Collection Data Types:

Lists- Introduction to python list, creating lists, Accessing list elements, List Operations (Concatenation, Repetition, membership), Modifying list, built-in list methods, Aliasing and Cloning lists, Sorting list elements, nested lists.

Tuples- Introduction to Python Tuples, Creating and Accessing tuple elements, Basic operations on tuples, Nested tuples, tuple methods, tuple unpacking. **Dictionaries-**Introduction to python Dictionaries, Creating , accessing and modifying Dictionaries, Dictionary methods, membership in Dictionaries, sorting dictionary elements, working with nested Dictionary. Converting between data types

Sets - Introduction to Sets, creating Sets, Accessing set elements, modifying sets, Set operations, Built-in Set methods **Files** - Files with built-in functions, Operations on files

Module-4

Object Oriented Programming:

Classes and Objects: Introduction to Object-Oriented Programming, Understanding Classes and Objects, Creating Classes in Python, The 'self' Variable, Types of Variables in a Class, Types of Methods in a Class, Namespace in Classes, Inner Classes, Passing Members Between Classes.

Inheritance and Polymorphism: Introduction to Inheritance, Types of Inheritance, Implementing Inheritance, Introduction to Polymorphism, Types of Polymorphism, Implementing Polymorphism, Operator Overloading, Inheritance and Polymorphism, Method Resolution Order (MRO).

Abstract classes and Interfaces: Introduction to Abstract Classes, Implementing Abstract Classes in Python, Introduction to Interfaces, Interfaces in Python, Abstract Classes vs. Interfaces.

Module-5

Python Module: creating user module, importing module, Creating Package, Modules – Random and Time. Other Useful modules in Python (Datetime ,Numpy, Pandas, os, sys)

Graphical User Interface Development (tkinter): Introduction, Components and Events, The root Component, Font and colors, working with containers, canvas, Frames. Widgets

PRACTICAL COMPONENT

SL. NO		PROGRAM NAME / DESCRIPTION
1		**OPERATORS**
	a	Read a list of numbers and write a program to check whether a particular element is present or not using membership operators.
	b	Read your name and age and write a program to display the year in which you will turn 100 years old
	c	Read radius and height of a cone and write a program to find the volume of a cone.
	d	Write a program to compute distance between two points taking input from the user (Hint: use Pythagorean theorem)
2		**CONTROL STRUCTURES**
	a	Read your email id and write a program to display the no of vowels, consonants, digits and white spaces in it using if…elif…else statement.
	b	Write a program to create and display a dictionary by storing the antonyms of words. Find the antonym of a particular word given by the user from the dictionary using while loop
	c	Write a Program to find the sum of a Series 1/1! + 2/2! + 3/3! + 4/4! +…….+ n/n!. (Input: n = 5, Output: 2.70833)
	d	In number theory, an abundant number or excessive number is a number for which the sum

		of its proper divisors is greater than the number itself. Write a program to find out, if the given number is abundant. (Input: 12, Sum of divisors of 12 = 1 + 2 + 3 + 4 + 6 = 16, sum of divisors 16 > original number 12)
3	**STRING**	
	a	Given a string, write a program to check if the string is symmetrical and palindrome or not. A string is said to be symmetrical if both the halves of the string are the same and a string is said to be a palindrome string if one half of the string is the reverse of the other half or if a string appears same when read forward or backward.
	b	Write a program to read a string and count the number of vowel letters and print all letters except 'e' and 's'.
	c	Write a program to read a line of text and remove the initial word from given text. (Hint: Use split() method, Input : India is my country. Output : is my country)
	d	Write a program to read a string and count how many times each letter appears. (Histogram).
4	**USER DEFINED FUNCTIONS**	
	a	A generator is a function that produces a sequence of results instead of a single value. Write a generator function for Fibonacci numbers up to n..
	b	Write a function merge_dict(dict1, dict2) to merge two Python dictionaries.
	c	Write a fact() function to compute the factorial of a given positive number
	d	Given a list of n elements, write a linear_search() function to search a given element x in a list
5	**BUILT IN FUNCTIONS**	
	a	Write a program to demonstrate the working of built-in statistical functions mean(), mode(), median() by importing statistics library.
	b	Write a program to demonstrate the working of built-in trignometric functions sin(), cos(), tan(), hypot(), degrees(), radians() by importing math module.
	c	Write a program to demonstrate the working of built-in Logarithmic and Power functions exp(), log(), log2(), log10(), pow() by importing math module.
	d	Write a program to demonstrate the working of built-in numeric functions ceil(), floor(), fabs(), factorial(), gcd() by importing math module.
6	**LIST**	
	a	Demonstrate a program that generates a list of 20 random numbers between 1 t0 100. i) Print the list ii) Print the average of the elements in the list Iii) print the largest and smallest values in the list iv)print how many even numbers are in the list.
	b	Write a program that removes anyrepeated items from a list so that each item appears at most once.
	c	Write a program to find sum of the numbers for the elements of the list by using reduce()?
	d	Write a program for map() function to double all the items in the list?
7	**CLASS AND OBJECTS**	
	a	Write a program to create a BankAccount class. Your class should support the following methods for i) Deposit ii) Withdraw iii) GetBalanace iv) PinChange
	b	Create a SavingsAccount class that behaves just like a BankAccount, but also has an interest rate and a method that increases the balance by the appropriate amount of interest (Hint:use Inheritance).
	c	Write a program to create an employee class and store the employee name, id, age, and salary using the constructor. Display the employee details by invoking employee_info()

		method and also using dictionary (__dict__).
	d	Access modifiers in Python are used to modify the default scope of variables. Write a program to demonstrate the 3 types of access modifiers: public, private and protected.
8		**TK INTERFACE**
	a	Write a python code to set background color and pic and draw a circle using turtle module
	b	Write a python code to set background color and pic and draw a square and fill the color using turtle module
	c	To implement a loan calculator using Tkintek
	d	To create a popup menu for arithmetic operations using Tkinter.
	e	To read and display an RGB color image and convert it into grayscale, negative and edge images.

TEXTBOOKS:

1. R Nageswara Rao, “Core Python Programming”, Dream tech Press, 2018 edition.
2. Grayson E. John, “Python and Tkinter Programming”, Manning Publications,1st edition,2000.
3. Eric Mattes “**PYTHON CRASH COURSE”,** Ahands-on, Project-based Introduction to Programming. 3rd Edition 2023, No starch press

REFERENCE BOOKS:

1. Lutz Ascher, “Learning Python”, O'Reilly, 4th edition,2009.
2. Chun J Wesley,“Core Python Applications Programming”, Pearson Education,3rd edition,2013.
3. Gries Paul et al., “Practical Programming: An introduction to Computer Science Using Python”, Pragmatic Bookshelf,3rd edition, 2018.
4. Downeyet Allen et.al, “Learning with Python: How to Think Like a Computer Scientist Dive into Python”2nd edition, 2002.

Weblinks and Video Lectures (e-Resources):

1. https://nptel.ac.in/noc/courses/noc19/SEM1/noc19-cs31/

4. **Syllabus Timeline**

S/L	Syllabus Timeline	Description
1	**Week 1-2 Basics of Python Programming**	Introduction to Python Programming: Overview of Python, syntax, basic data types, variables, operators, expressions, statements, and control flow statements. Control Flow Statements: Detailed exploration of sequence, selection (if statements), and iteration (loops). Introduction to control flow modifiers.
2	**Week 3-5: Functions in Python**	Functions in Python : Defining and calling functions, understanding parameters and return values. Introduction to void functions. Recursive functions, default parameters, and exception handling. Strings: Basics of string operations, methods, formatting, slicing, and indexing.
3	**Week 6-8: Collection Data Types**	List: Creating lists using range(), updating elements, concatenation, and repetition. Membership, aliasing, cloning, sorting, and working with nested lists. Tuples and Sets: Creating, accessing, and manipulating tuples and sets. Understanding tuple operations, nested tuples, and set operations. Dictionaries and File Handling: Dictionary operations and methods, sorting dictionary elements, converting lists to dictionaries, and basic file handling with built-in functions.
4	**Week 9-11: Object-**	Introduction to classes, objects, constructors, and methods. Understanding self and namespaces. Inheritance, polymorphism, method resolution order (MRO),

	Oriented Programming	operator overloading, and method overriding.
5	**Week 12-13: Graphical User Interface Development**	Basics of GUI design using tkinter, creating windows and frames. Introduction to widgets and their properties. Handling events, working with advanced widgets, refining the GUI. Python Modules and Libraries: Creating, importing, and using modules and packages. Overview of practical projects integrating learned skills.

5. **Teaching-Learning Process Strategies**

S/L	TLP Strategies:	Description
1	**Lecture Method**	Deliver structured lectures on Python programming basics, including data types, variables, control structures, and functions. Use clear explanations and real-life examples to reinforce learning..
2	**Video/Animation**	Utilize videos and animations to demonstrate Python programming concepts, such as control flow, functions, and GUI development. Visual aids will enhance understanding of coding and programming concepts.
3	**Collaborative Learning**	Encourage group projects and collaborative activities to apply Python programming concepts, design solutions, and solve problems together. This will improve teamwork and practical coding skills.
4	**Hands-On Programming**	Conduct hands-on coding sessions where students actively write and test Python code. This will solidify understanding of concepts and improve practical coding skills.
5	**Interactive Tutorials**	Use interactive tutorials and coding exercises to provide immediate feedback on Python programming tasks. This helps reinforce learning and addresses areas of difficulty in real-time.
6	**Peer Review and Feedback**	Implement peer review sessions where students evaluate each other’s code and provide constructive feedback. This will enhance learning through collaboration and improve code quality.
7	**Quizzes and Assessments**	Conduct regular quizzes and assessments to evaluate students' understanding of Python programming concepts and provide targeted feedback. This will help in tracking progress and identifying areas for improvement.
8	**Guest Lectures/Workshops**	Invite industry experts to deliver guest lectures or conduct workshops on emerging IoT technologies and trends. This provides students with insights into current industry practices and innovations.

6. **Assessment Details (both CIE and SEE)**

Continuous Internal Evaluation:

CIE Split up for Integrated Professional Core Course (IPC)

The minimum CIE marks requirement is 50% of maximum marks in each component.

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Theory (A)	Internal Assessment-Tests (A)	2	**60%**	**15**	**7.5**
	Assignments/Quiz/Activity (B)	2	**40%**	**10**	**05**
	Total Marks		**100%**	**25**	**12.5**
Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Laboratory(B)	Record Writing	Continuous	**60%**	15	7.5
	Test at the end of the semester	1	**40%**	10	05
	Total Marks		**100%**	**25**	**12.5**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

1. Question paper pattern will be ten questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understand Python Basics**	Understand the fundamental concepts of Python programming, including data types, variables, control structures, and basic syntax.
2	**Master Functions and Code Reusability**	Develop proficiency in creating and using functions to enhance code reusability, readability, and maintainability. Learn about recursion and exception handling.
3	**Utilize Storage Structures and Data Types**	Gain a deep understanding of various data structures and collection types in Python, including strings, lists, tuples, sets, and dictionaries. Learn to manipulate and operate on these data types effectively.
4	**Implement Object-Oriented Programming**	Apply object-oriented programming principles in Python, including creating classes, objects, inheritance, and polymorphism to design and implement robust and maintainable code.
5	**Develop Graphical User Interfaces**	Learn to design and develop graphical user interfaces (GUIs) using tkinter, including creating windows, frames, and interactive widgets.
6	**Apply Python Modules and Libraries**	Utilize various Python modules and libraries for practical applications, including data science, web development, automation, and more. Learn to create and manage Python modules and packages.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23MCA204.1	Present a comprehensive understanding of the fundamentals of Python programming, including data types, variables, control structures, and basic syntax.
M23MCA204.2	Apply Python functions to write reusable code, handle exceptions, and enhance code readability and maintainability
M23MCA204.3	Analyze various data structures and collection types in Python, such as strings, lists, tuples, sets, and dictionaries, to manage and manipulate data effectively
M23MCA204.4	Evaluate object-oriented programming principles in Python, such as classes, inheritance, and polymorphism, to design effective and well-organized code.
M23MCA204.5	Develop practical applications by designing and implementing graphical user interfaces (GUIs) using Python libraries, such as tkinter.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA204.1	3	-	-	-	-	-	-	
M23MCA204.2	3	-	-	-	-	-	-	2
M23MCA204.3	-	3	-	-	-	-		
M23MCA204.4	-	-	3	3	-	-	-	-

M23MCA204.5	-	-	-	3	-	-	-	-
M23MCA204	3	3	3	3	-	-	-	2

9. **Assessment Plan**

Continuous Internal Evaluation (CIE)

	CO1	**CO2**	**CO3**	**CO4**	**CO5**	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	**CO2**	**CO3**	**CO4**	**CO5**	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

10. **Future with this Subject:**

Embracing Advanced Tools:

New Libraries and Tools: Update the course to include cutting-edge Python libraries such as Dask and PySpark for handling large data sets. Introduce students to modern development environments like PyCharm and Visual Studio Code to enhance their coding efficiency.

Data Science and Machine Learning:

Advanced Data Analysis: Expand the curriculum to cover more in-depth data analysis using libraries like Pandas and NumPy. Engage students in hands-on projects with real-world data to build practical skills.

Introduction to Machine Learning: Add a new section that introduces basic machine learning concepts using scikit-learn, TensorFlow, and Keras. This will help students understand how to create and apply predictive models.

Exploring Web Development and Automation:

Building Web Applications: Enhance the GUI module to include web development using frameworks like Flask and Django. This will allow students to create interactive web applications and learn about web technologies.

Automating Tasks: Introduce a new module focused on automating everyday tasks with Python. Cover topics such as web scraping using BeautifulSoup and automated testing with Selenium to streamline workflows.

2nd Semester	PROFESSIONAL ELECTIVE 1(PE) SOFTWARE DEVELOPMENT & DEVOPS	M23MCA205A

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming skills**	Ability to write and understand code, including variables, loops, conditional statements, and functions/methods
2	**Understanding of Software Development Life Cycle**	Ability to comprehend the sequential and iterative nature of software development processes.
3	**Foundational Knowledge of Agile Principles**	Ability to grasp concepts such as iterative development, continuous feedback, and adaptive planning
4	**Basic Understanding of Cloud Computing**	Familiarity with cloud service models (IaaS, PaaS, SaaS) and cloud deployment models (public, private, hybrid).
5	**Basic Linux/Unix Command Line Skills**	Ability to navigate the file system, manipulate files and directories, and execute commands in a terminal environment.
6	**Problem Solving and Collaboration Skills**	Ability to work collaboratively in a team environment and communicate effectively with colleagues and stakeholders.

2. Competencies

S/L	Competency	KSA Description
1	**Understanding of Agile Principles and Practices**	**Knowledge:** • Knowledge of Agile methodologies such as Scrum, Kanban, and Extreme Programming (XP). **Skills:** • Understanding of Agile principles, values, and the Agile Manifesto. Proficiency in utilizing Quine-McCluskey minimization techniques **Attitudes:** • Appreciation for the Familiarity with Agile frameworks and their application in software development.
2	**Knowledge of DevOps Concepts and Practices**	**Knowledge:** • Understanding of DevOps principles, including collaboration, automation, measurement, and sharing (CAMS). **Skills:** • Knowledge of DevOps practices such as continuous integration, continuous delivery, infrastructure as code, and automated testing **Attitudes:** • Appreciation for the awareness of DevOps tools and technologies used for deployment, monitoring, and orchestration.
3	**Technical Knowledge**	**Knowledge:** • Proficiency in programming languages commonly used in software development (e.g., Java, Python, JavaScript). **Skills:** • Understanding of version control systems (e.g., Git) and their role in collaborative development. **Attitudes**

		• Valuing the importance of Knowledge of containerization technologies(e.g., Docker) and container orchestration platforms.

3. Syllabus

SOFTWARE DEVELOPMENT AND DEVOPS
SEMESTER – II

Course Code	**M23MCA205A**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives:

1. The importance of the software development process.
2. Demonstrate the workflow of Automating process.
3. The development of a software using Agile method
4. Illustrate with case study, the importance of DevOps.
5. Essential software development activities.

Module -1	
Introduction Defining the Software Development Process: Goals of defining the software development process, Why is defining software development process important?, Where do I start?, Explaining the software development lifecycle, System versus software development lifecycle defining requirements, Managing complexity and change, Validity of requirements, Testing requirements, Fundamental requirements, Non-fundaments requirements, Epics and stories, Planning for changing requirements, workflow of defining requirements, Test-driven developments, Designing systems, Software development, Testing, Testing the applications, Testing the process itself, Continuous Integration, Continuous Delivery and Deployment, Defining phases of the lifecycle, Documentation required , DevOps, Communicating with all stakeholders,Production support, Maintenance and bugfixes, Lifecycle in the beginning , Maintenance of the lifecycle, Creating the knowledge base. **(Chapter 1 & 2).**	L1, L2, L3
Module -2	
Agile Application Lifecycle Management: Goals of Agile Application Lifecycle Management, Why Is Agile ALM Important? Where Do I Start? Understanding the Paradigm Shift, Rapid Iterative Development, Remember RAD?, Focuson12 Agile Principles, Agile Manifesto, Fixed Time box Sprints, Customer Collaboration, Requirements and Documentation**.** **(Chapter 3).**	L1, L2, L3
Module -3	
Automating the Agile ALM: Goals of Automating the Agile ALM, Why Automating the ALMIs Important, Where Do I Start? Tools, Do Tools Matter? Process over Tools, Understanding Tools in the Scope of ALM, Staying Tools Agnostic, Commercial versus Open Source, What Do I Do Today?, Automating the Workflow, Process Modelling Automation, Managing the Lifecycle with ALM, Broad Scope of ALM Tools ,Achieving Seamless Integration ,Managing Requirements of the ALM, Creating Epics and Stories, Systems and Driven Development, Environment Management, Gold Copies, Supporting the CMDB, Driving DevOps ,Supporting Operations ,Help Desk ,Service Desk ,Incident Management , Problem Escalation ,Project Management, Planning the PMO ,Planning for Implementation, Evaluating and Selecting the Right Tools, Defining the Use Case, Training Is Essential, Vendor Relationships, Keeping Tools Current. **(Chapter 7).**	L1, L2, L3
Module -4	
Change Management: Why Is Change Management Important?, Traceability for Compliance, Assess and Manage Risk, Communication, Change in Application Lifecycle Management, The Change Ecosystem, QA and Testing, Establishing the Command Center, The Change Management Process, Preapproved Changes, Establishing the Change Management Function, Change Control Topology, Coordinating across the Platform- enterprise, Specialized Change Control, Vendor	L1, L2, L3

Change Control, SaaS Change Control, Continuous Process Improvement, IT Operations: Why Is IT Operations Important?, Monitoring the Environment, Production Support, Help Desk, IT Process Automation, Workflow Automation, Escalation. **(Chapter 10 & 11).**	
Module -5	
DevOps: Goals of DevOps, Why Is DevOps Important? Where Do I Start? How Do I Implement DevOps? Developers and Operations Conflicts, Developers and Operations Collaboration, Need for Rapid Change, Knowledge Management, the Cross-Functional Team, Is DevOps Agile? The DevOps Ecosystem, Moving the Process Upstream, Left-Shift, Right-Shift, DevOp sinDev, DevOps as Development, Deployment Pipeline, Dependency Control, Configuration Control, Configuration Audits, QA and DevOps, Information Security, Infrastructure as Code, Taming Complexity, Automate Everything, Disaster Recovery and Business Continuity, Continuous Process Improvement. **(Chapter 12).**	L1, L2, L3
Text Books: 1. BobAiello and LeslieSachs, "Agile Application Life cycle Management Using DevOps to Drive Process Improvement", Addison Wesly, First printing, 2016. Reference Books: 1. Roger S, "Software Engineering-A Practitioner's Approach", seventh edition, Pressman, 2010. 2. Roger Pressman, Ian sommerville, "Software Engineering", Pearson,9th edition,2010. 3. HansVanVliet, "Software Engineering: Principles and Practices", Wiley,2008.	

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction Defining the Software Development Process**	• Knowledge of Agile methodologies such as Scrum, Kanban, and Extreme Programming (XP). • Software Development Methodologies Software Development Lifecycle (SDLC) • Understanding of Agile principles, values, and the Agile Manifesto. • Proficiency in utilizing Quine-McCluskey minimization techniques.
2	**Week 3-4: Agile Application Life cycle Management**	• Knowledge of DevOps Concepts and Practices • Understanding of DevOps principles, including collaboration, automation, measurement, and sharing (CAMS). • Effective Communication and Collaboration • Agile Planning and Execution
3	**Week 5-6: Automating the Agile ALM**	• Technical Knowledge • Understanding of Automation Tools and Technologies • Automation Scripting and Programming • Configuration and Management of Automation Tools
4	**Week 7-8: Change Management**	• Understanding of Change Management Principles • Models Knowledge of Organizational Behavior and Culture • Stakeholder Engagement and Communication Change Planning and Implementation
5	**Week 9-10: DevOps**	• Understanding of DevOps Principles and Culture • Knowledge of DevOps Tools and Technologies • Automation and Scripting • Collaboration and Communication
6	**Week 11-12: DevOps as Development**	• Understanding of DevOps Development • Apply learned concepts and competencies to real-world scenarios. • Hands-on practice with programming assignments

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Using traditional lecture methods and ICT as and when needed.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance learning.
3	**Collaborative Learning**	Encourage collaborative learning approaches for peer learning.
4	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application.
5	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
6	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(two) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understand Agile Principles and Methodologies**	Students will grasp the fundamental concepts of roles, responsibilities, and ceremonies in Agile development
2	**Learn DevOps Concepts and Practices**	Students will learn to Identify the key components of a DevOps culture and how they contribute to organizational success and also to understand the DevOps tool chain and its role in automating the software delivery pipeline
3	**Explore Agile Project Management**	Students will become proficient in Create and manage Agile artifacts such as user stories, product backlogs, and sprint plans.
4	**Implement Continuous Integration and Continuous Delivery (CI/CD)**	Through hands-on projects, students will apply their knowledge of Integrate version control, automated testing, and deployment automation tools into CI/CD workflows.
5	**Practice Collaboration and Communication**	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.

	Skills	
6	**Apply Agile and DevOps Principles in Real-world Scenarios**	Students will understand to address the challenges and adapt Agile and DevOps methodologies to suit the needs of specific projects, teams, and organizational contexts. Reflect on experiences and lessons learned to continuously improve Agile and DevOps implementation.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA205A .1	Understand and apply the concepts of DevOps.
M23MCA205A.2	Apply the concepts of DevOps for a given Scenario
M23MCA205A.3	Design a software system, component or process to meet desired needs within realistic constraints

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA205A .1	3	-	-	-	-	-	-	-
M23MCA205A.2	-	3	-	-	-	-	-	-
M23MCA205A.3	-	-	3	-	-	-	-	-
M23MCA205A	3	3	3	-	-	-	-	-

Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	Total
Module 1	10			10
Module 2	10	10		20
Module 3		5		5
Module 4			10	10
Module 5			5	5
Total	20	15	15	50

Semester End Examination (SEE)

	CO1	CO2	CO3	Total
Module 1	20			20
Module 2	20	20		40
Module 3		10		10
Module 4			20	20
Module 5			10	10
total	40	30	30	100

10. Future with this Subject

The future of Agile Software Development and DevOps is promising, with continued growth and adoption expected in the coming years. Here are some key trends and developments shaping the future of these subjects:

- **Integration of Agile and DevOps Practices:** Organizations are increasingly recognizing the complementary nature of Agile and DevOps methodologies and seeking to integrate them into a unified approach for software delivery. Agile practices focus on iterative development, customer collaboration, and adaptability, while DevOps emphasizes automation, collaboration, and continuous delivery. Integrating these practices enables organizations to accelerate software delivery while maintaining quality and reliability.

- **Shift towards Value Stream Management (VSM**): Value Stream Management (VSM) is emerging as a strategic approach to optimize the end-to-end software delivery process, from ideation to deployment and beyond. VSM focuses on identifying and eliminating bottlenecks, reducing cycle times, and maximizing value delivery to customers.
- **Emphasis on DevSecOps and Continuous Security:** With the growing importance of cybersecurity and data privacy, organizations are prioritizing security throughout the software development lifecycle. DevSecOps integrates security practices into the DevOps pipeline, enabling automated security testing, vulnerability scanning, and compliance checks.
- **Adoption of No-Code/Low-Code Development Platforms:** No-code/low-code development platforms are gaining popularity as organizations seek to accelerate application development and empower citizen developers. These platforms enable rapid prototyping, visual development, and automation of repetitive tasks, reducing the need for manual coding and shortening time-to-market..
- **Expansion of AI and Machine Learning in Software Development:** Artificial Intelligence (AI) and Machine Learning (ML) technologies are increasingly being integrated into Agile and DevOps processes to automate tasks, improve decision making, and enhance predictive analytics. AI/ML algorithms can analyze large datasets to identify patterns, predict failures, optimize resource allocation, and provide insights for continuous improvement.

Focus on Continuous Learning and Improvement: Continuous learning and improvement remain fundamental principles of Agile and DevOps cultures. Organizations are investing in training, coaching, and up skillin programs to build capabilities and foster a culture of innovation and adapt ability.

2nd Semester	**PROFESSIONAL ELECTIVE- I (PE)** **DATA WAREHOUSE AND DATA MINING**	**M23MCA205B**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming Skills**	Proficiency in at least one programming language, such as Python, Java, or SQL, is essential for manipulating data and understanding the underlying algorithms.
2	**Database Knowledge**	Understanding relational database concepts, including SQL queries, schema design, normalization, and transactions
3	**Statistics and Probability**	Basic knowledge of statistics and probability is crucial for understanding data analysis and mining techniques.
4	**Data Structures and Algorithms**	Familiarity with fundamental data structures (like arrays, lists, trees, and graphs) and algorithms is important for efficient data processing and manipulation.
5	**Mathematics**	A good grasp of linear algebra and calculus can be beneficial, especially for understanding some of the mathematical foundations of data mining algorithms.
6	**Basic Understanding of Data Warehousing Concepts**	While not always required, having an introductory knowledge of data warehousing concepts like ETL (Extract, Transform, Load) processes, OLAP (Online Analytical Processing), and data warehouse architecture can be helpful.
7	**Problem-Solving Skills**	Strong analytical and problem-solving skills to tackle complex data challenges.
8	**Introductory Data Science or Data Analytics**	Basic courses in data science or data analytics can provide a good foundation, as they often cover essential concepts that will be expanded upon in data warehousing and data mining courses.

2. Competencies

S/L	Competency	KSA Description
1	**Data Warehousing Concepts**	**Knowledge** • Understanding of ETL (Extract, Transform, Load) processes. Knowledge of data warehouse architecture, OLAP (Online Analytical Processing), and data warehousing models. **Skills** • Proficiency in SQL and database querying. Ability to write and optimize SQL queries for data extraction and manipulation. **Attitude** • Curiosity and eagerness to learn about data warehousing. A genuine interest in exploring data and discovering hidden patterns or insights.
2	**Building a Data Warehousing**	**Knowledge** • Understanding of data warehouse planning, physical structure, and conceptual modeling. Knowledge of multidimensional data models, OLAP servers, and data warehousing schemas. **Skills** • Ability to design and implement data warehouses. Skills in querying multidimensional data Models. **Attitude** • Detail-oriented approach to designing and implementing data warehousing solutions. Willingness to experiment with different data modeling techniques and OLAP operations.

3	**Data Mining Techniques**	**Knowledge** • Understanding of various data mining methods such as classification, clustering, regression, association rule mining, and anomaly detection. Familiarity with data preprocessing techniques including cleaning, reduction, and transformation. **Skills** • Proficiency in programming languages such as Python or R for data manipulation and analysis. Expertise in designing and implementing ETL processes for data integration. Ability to apply mining algorithms like Apriori and FP-Tree for frequent pattern mining. **Attitude** • Careful and meticulous approach to data handling to avoid errors and ensure accuracy. Willingness to tackle complex and sometimes frustrating data challenges without giving up.
4	**Classification Techniques**	**Knowledge** • Understanding of classification methods including decision trees, Bayes methods, rule-based classification, and support vector machines. Familiarity with other classification techniques such as genetic algorithms, rough set approach, and fuzzy set approach. **Skills** • Ability to develop and evaluate classification models. Skills in implementing various classification algorithms and selecting the best model based on evaluation metrics. **Attitude** • Analytical mindset towards understanding and applying classification methods. Curiosity in exploring and mastering advanced classification techniques.
5	**Cluster Analysis**	**Knowledge** • Understanding of clustering methods including partitioning, hierarchical, density-based, and grid-based methods. Knowledge of clustering evaluation techniques and metrics **Skills** • Proficiency in implementing and tuning clustering algorithms. Ability to evaluate clustering results and select the most appropriate clustering method for different datasets. **Attitude** • Detail-oriented approach to implementing and evaluating clustering techniques. Experimental mindset in exploring various clustering methods to achieve optimal results.

3. Syllabus

<table>
<tr><td colspan="5"><b>DATA WAREHOUSE AND DATA MINING</b>
<b>SEMESTER – II</b></td></tr>
<tr><td>Course Code</td><td><b>M23MCA205B</b></td><td>CIE Marks</td><td colspan="2"><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week (L: T: P: S)</td><td><b>(3:0:0:0)</b></td><td>SEE Marks</td><td colspan="2"><b>50</b></td></tr>
<tr><td>Total Number of Lecture Hours</td><td><b>40 hours</b></td><td>Total Marks</td><td colspan="2"><b>100</b></td></tr>
<tr><td>Credits</td><td><b>03</b></td><td>Exam Hours</td><td colspan="2"><b>03</b></td></tr>
<tr><td colspan="5"><b>Course objectives:</b>
1. To be familiar with mathematical foundations of data mining tools
2. To implement classical models and algorithms in data warehouses and data mining
3. To analyze patterns that can be discovered by association rule mining, classification and clustering.
4. To develop skills in selecting the appropriate data mining algorithm for solving problems</td></tr>
<tr><td colspan="5"><b>Module 1</b></td></tr>
<tr><td colspan="4"><b>Introduction</b>
Data warehousing, data warehouse Description, Three-layer Architecture: Conceptual view. Data Warehousing: concepts & mechanisms-Introduction, Need for Developing Data warehouse, What</td><td>L1,L2</td></tr>
</table>

is a Data Warehouse? Why separate data Warehouse? Data warehouse systems, Data warehouse Components, administration and management Tools, Data Mart, The difference between OLTP and Data Warehousing, Decision Support and OLTP, Data processing and models, Using Data Warehousing in strategic Decision Making.	
Module 2	
Building a Data Warehousing Introduction, Planning a Data warehouse, creating and Maintaining a warehouse, Physical Structure of data warehouse, conceptual Modeling of Data warehouse, Multidimensional Data model, OLAP servers, implementing a Warehouse, Database System vs Data warehouse. Introduction, OLAP-OLAP server, OLAP by example, Typical OLAP operations, Query model for querying Multidimensional databases.	L2,L3
Module 3	
Introduction to Data Mining Why data Mining? What is Data Mining? What Kind of data can be mined? What kinds of patterns can be mined? Which technologies are used, Which Kinds of Applications are targeted. Data Preprocessing: An overview, Data Cleaning, data Reduction, Data Transformation .Mining frequent patterns, Associations: Market Basket Analysis, Frequent itemsets, closed itemsets and Association rules, Frequent Itemset Mining Methods, Apriori and FP-Tree growth Algorithm.	L2,L3, L4
Module 4	
Classification Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Model Evaluation and Selection, Support vector machine, K-Nearest-Neighbour Classifiers, other Classification Methods: Generic Algorithms, Rough Set Approach, Fuzzy Set Approach	L3,L4
Module 5	
Cluster Analysis Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Method, Evaluation of Clustering.	L3,L4

Text Books:

1. Gajendra Sharma, Data Mining, Data Warehousing and OLAP, Katson Books, 2019
2. Jiawei Han and MichelineKamber, Data Mining - Concepts and Techniques, 2nd Edition,Morgan Kaufmann Publisher,

Reference Books:

1. Alex Berson and Stephen J smith , Data Warehousing, Data Mining, & OLAP, Tata Mcgraw-Hill, 2018.
2. Paulraj Ponnaiah, Data Warehousing fundamentals for IT professionals, wiley student publishers,second edition, 2014.
3. Ralph Kimball, MargyRoss,The data warehouse toolkit, third edition, wiley publishers,2012.

Journals/Magazines:

1. Data Mining and Knowledge –Springer publisher.
2. International Journal of Data Structures-STM Journals publisher.
3. Journal of Data Mining and Management – MAT Journals publisher.
4. International Journal of Data Mining, Modelling and Management-INDER Science Journals

Web/Digital resources:

1. https://www.coursera.org/specializations/data-warehousing.
2. www.knowledge-management-tools.net/data-warehousing.htm.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-3** **Data warehousing, data warehouse**	Data warehousing, data warehouse Description, Three-layer Architecture: Conceptual view. Data Warehousing: concepts & mechanisms-Introduction, Data Mart, The difference between OLTP

	Description	Using Data Warehousing in strategic Decision Making.
2	**Week 4-6** **Building a Data Warehousing**	Introduction, planning a Data warehouse, creating and maintaining a warehouse, OLAP-OLAP server, OLAP by example, data warehousing modeling: Data Cube and OLAP Data Cube Schemas for Multidimensional Data Models, dimensions.
3	**Week 7-9** **Data Mining**	Data Mining, Kind of data can be mined, technologies are used, Which Kinds of Applications are targeted. Data Preprocessing: An overview, Data Cleaning, data Reduction, Data Transformation. Mining.
4	**Week 10-12** **Classification**	Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Model Evaluation and Selection, Support vector machine, K-Nearest-Neighbor Classifiers, other Classification Methods
5	**Week 12-14** **Cluster Analysis**	Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Method, Evaluation of Clustering

5. Teaching-Learning Process Strategies

S/L	TLP Strategies	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions**	Pose HOTS questions to stimulate critical thinking related to each competency
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

	Components	Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a

maximum of 3 sub questions), may have mix of topics under that module if necessary.

3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understand Data Storage**	Learn how to collect, store, and manage large amounts of data efficiently using data warehouses. Be familiar with mathematical foundations of data mining tools.
2	**Extract Insights**	Gain the skills to analyze data and discover useful patterns and relationships through data mining techniques. Implement classical models and algorithms in data warehouses and data mining
3	**Improve Decision Making**	Develop the ability to use data to make informed business decisions and solve real-world problems. Discover interesting patterns using association rule mining, classification and clustering
4	**Hands-On Practice**	Get practical experience with tools and techniques used in the industry to handle and analyze big data. Develop skill in selecting the appropriate data mining algorithm for solving practical problems

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA205B.1	Understand the fundamentals of Data Warehousing, Conceptual modeling of data warehouses, multidimensional data model, Data mining, KDD Process.
M23MCA205B.2	Apply OLAP and Multidimensional Analysis, Clustering Methods, various classification techniques and algorithms like Apriori and FP-Tree for pattern recognition and association rule mining to real-world data.
M23MCA205B.3	Analyze the frequent patterns using association analysis algorithms, classification and clustering outcomes using internal and external evaluation metrics to ensure effective data segmentation.
M23MCA205B.4	Evaluate Model Performance: Use metrics such as accuracy, precision, recall, and F1 score to compare and contrast the various classifiers

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA205B.1	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205B.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205B.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205B.4	**-**	**-**	**2**	**-**	**-**	**-**	**-**	**-**
M23MCA205B	**2.5**	**3**	**2**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**10**				**10**
Module 3		**10**			**10**
Module 4			**10**		**10**
Module 5				**10**	**10**
Total	**20**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**

Module 1	**20**				**20**
Module 2	**20**				**20**
Module 3		**20**			**20**
Module 4			**20**		**20**
Module 5				**20**	**20**
Total	**40**	**20**	**20**	**20**	**100**

10. Future with this Subject

The future of data warehousing and data mining courses looks very promising and continues to evolve with technological advancements and industry needs.

13. **Integration with Big Data Technologies**: Courses will increasingly cover big data platforms like Hadoop and Spark, teaching how to manage and analyze massive datasets efficiently.
14. **Focus on Real-Time Data Processing**: There will be a growing emphasis on real-time data warehousing and mining, addressing the need for immediate data insights and decision-making.
15. **Advanced Analytics and AI**: The curriculum will incorporate advanced analytics, machine learning, and AI techniques, enabling more sophisticated data mining and predictive analytics.
16. **Cloud-Based Solutions**: With the rise of cloud computing, courses will focus on cloud-based data warehousing solutions like Amazon Redshift, Google BigQuery, and Microsoft Azure Synapse.
17. **Data Privacy and Ethics**: As data privacy concerns grow, there will be a stronger emphasis on ethical data handling, privacy laws, and secure data management practices.
18. **Interdisciplinary Applications**: Data warehousing and mining will be integrated with various domains such as healthcare, finance, marketing, and more, showing how these skills apply to different industries.
19. **Automation and Tools**: Students will learn about automation in ETL processes, data integration tools, and the latest software for data analysis and visualization.
20. **Practical, Hands-On Learning**: The focus will be on practical, hands-on experiences, with real-world projects and case studies to prepare students for industry challenges.
21. **Collaboration with Industry**: Increased collaboration with industry partners to ensure that the course content is aligned with current market needs and trends.

2nd Semester	**PROFESSIONAL ELECTIVE 1(PE)** **UNIX AND SHELL PROGRAMMING**	**M23MCA205C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic Computer Skills**	Familiarity with using a computer, navigating the file system, managing files and directories, and using the command line interface (CLI) is essential.
2.	**Understanding of Operating Systems**	Basic knowledge of how operating systems work, including processes, memory management, file systems, and user permissions.
3.	**Familiarity with Command Line Interface (CLI)**	Understanding of how to navigate directories, list files, create directories, copy/move files, and execute commands using the command line interface.
4.	**Programming Fundamentals**	Basic understanding of programming concepts like variables, data types, loops, conditionals, functions, and control structures. This will help in understanding shell scripting.
5.	**Text Editing Skills**	Proficiency in using a text editor, as shell scripts are essentially text files containing commands.
6.	**Problem-Solving Skills**	Ability to analyze problems, break them down into smaller components, and devise solutions. Shell scripting often involves solving various problems efficiently.

2. Competencies

S/L	Competency	KSA Description
1.	**Proficiency in Command Line Interface**	**Knowledge:** **22.** Understand the fundamental of Command line Interface **Skills:** • Efficient file manipulation, text processing, and system administrations. **Attitudes:** • Be comfortable with command line interface
2.	**Shell Scripting**	**Knowledge:** • Understanding shell script writing. **Skills:** • Writing shell scripts to automate tasks, create custom utilities, and streamline workflows, **Attitudes:** • Confident in writing shell scripts.
3.	**System Administration Skills**	**Knowledge:** • Understanding system administration. **Skills:** • User management, file permissions, process management, and system monitoring. **Attitudes:** • Confident in managing UNIX/Linux-based systems
4.	**Text Processing and Manipulation**	**Knowledge:** • Understanding basic text processing and Manipulation. **Skills:** • Manipulate and process text using command line tools and shell scripting, including tasks such as searching, filtering, sorting, and transforming text data. **Attitudes:** • Comfortable in managing text.

3. Syllabus

UNIX AND SHELL PROGRAMMING
SEMESTER – II

Course Code	**M23MCA205C**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course Learning objectives:

The main objectives of this course are to:

- Understand the features, architecture of UNIX and its commands.
- Discuss different UNIX files, attributes and permissions.
- Discuss filter programs and regular expressions.
- Familiarize with advanced filters

Module -1	
UNIX Architecture and Command Usage: UNIX Architecture, Features of UNIX, Internal and External Commands. **General-Purpose Utilities**: cal, date, echo, printf, bc, passwd, who, uname, tty, stty. **The File System**: The Parent-Child Relationship, the HOME variable, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, ls: Listing Directory Contents.	L1, L2, L3
Module -2	
Handling Ordinary Files: cat, cp, rm, mv, more, file, wc, od, cmp, comm, diff, Basic File Attributes: ls –l, the –d option, file ownership, file permissions, chmod, directory permissions, changing file ownership. **More File Attributes**: File Systems and Inodes, Hard Links, Symbolic Links and ln, umask Modification and Access Times	L1, L2, L3
Module -3	
Simple Filters: The sample database, paginating files, head, tail, cut, paste, sort, uniq, tr. **Filters using Regular Expression**: grep: Searching for a pattern, Basic Regular Expression (BRE), egrep: Extended Regular Expression.	L1, L2, L3
Module -4	
Essential Shell Programming Part I: Shell Scripts, read, Using command line arguments, exit and exit status of command, the logical operators && and || - conditional execution, the if conditional, using test and [ ] to evaluate expressions, the case conditional, expr, $0, while, for, set and shift, trap: Interrupting a program	L1, L2, L3
Module -5	
awk: An Advanced Filter:Simple awk filtering, Splitting a Line into Fields, printf: Formatting Output, Variables and Expressions, Comparison Operators, Number Processing, Built-in Variable, Arrays, Functions, Control Flow, Looping with for and while	L1, L2, L3

Text Book(s)

1. UNIX – Concepts and Applications, Sumitabha Das, 4th Edition, McGraw Hill, 2017.

Reference Books

1. UNIX and SHELL Programming, Behrouz A Forouzan and Richard F Gilberg, India. Edition, Cengage Learning, Third Reprint 2008
2. UNIX – The Complete Reference, Kenneth Rosen et al, 2nd Edition, Tata McGraw Hill Fourth Reprint 2008
3. Your UNIX: The Ultimate Guide, Sumitabha Das, McGraw Hill, 2001.
4. Introduction to UNIX and Shell Programming, M G Venkateshmurthy, Pearson Edition.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: UNIX Architecture and Command Usage**	Understand UNIX Operating System Architecture. Acquire the Knowledge of UNIX command and its usage. Understand the file system.
2	**Week 3-4: Handling Ordinary Files**	Impart the knowledge of Command Line Interface. Different commands for handling files. Able to write shell scripts for handing files.
3	**Week 5-6: Simple Filters and Filters using Regular Expression**	Understand and apply simple Filters and Regular Expressions for solving various problems. Develop scripts for handling regular expressions
4	**Week 7-8: Essential Shell Programming**	Acquire the Knowledge: UNIX data types, operators, if conditional Statement and looping statements, etc. Use various UNIX features for developing scripts.
5	**Week 9-10: awk- An Advanced Filters**	Understand the importance of Advanced filters. Develop shell scripts using advanced filters.
6	**Week 11-12: Integration and Practical Applications**	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with programming assignments.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Live Demonstration**	Develop and run Shell scripts in the classroom.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Programming Assignments**	Assign programming tasks to improve the practical skills.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.

3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding the architecture of Linux operating System**	Students will understand the Linux operating system Architecture.
2	**Analyze the working of various Linus Commands**	Students will be able to analyze the working of various Linux commands by executing commands.
3	**Develop a Shell Script**	To create programs in the Linux environment using Linux utilities and commands.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA205C.1	Understand the fundamental concepts of UNIX Operating system and analyze working of various commands.
M23MCA205C.2	Apply various filters to solve variety of applications.
M23MCA205C.3	Develop Regular expressions for pattern matching.
M23MCA205C.4	Develop various shell scripts for performing various operations on Linux Operating System and use awk advanced filters.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA205C.1	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205C.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205C.3	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA205C.4	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA205C	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**	**10**			**20**
Module 2	**5**				**5**
Module 3		**5**			**5**
Module 4			**10**		**10**
Module 5				**10**	**10**
Total	**15**	**15**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**	**20**			**40**
Module 2	**10**				**10**
Module 3		**10**			**10**
Module 4			**20**		**20**
Module 5				**20**	**20**
Total	**30**	**30**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "UNIX and Shell Programming" course in the third semester of the B.E (Computer Science & Engineering Branches) program places an important role for learning several future courses in the undergraduate program. This subject is very important for conducting many laboratory subjects such as Analysis and Design of Algorithm, Database Management System, Data Structures, Python programming, etc.

Here are some notable contributions:

32. **Internet of Things (IoT) and Embedded Systems:** Many IoT devices and embedded systems run on UNIX-like operating systems or utilize shell scripts for managing system tasks. Understanding UNIX and shell programming is beneficial for developers working on IoT devices, embedded systems, and firmware development.
33. **Education and Training:** UNIX and shell programming concepts are often taught in computer science and information technology curricula as foundational skills. Aspiring software engineers, system administrators, and IT professionals continue to learn UNIX and shell programming to build a strong technical foundation.
34. **Data Processing and Analysis:** UNIX tools and shell scripting are commonly used for data processing, manipulation, and analysis tasks. As data continues to grow in volume and complexity, the ability to efficiently process and analyze data using command line tools and shell scripts remains relevant for data scientists, analysts, and researchers.

2nd Semester	**PROFESSIONAL ELECTIVE 1 (PE)** **DATA SCIENCE**	**M23MCA205D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Mathematics:**	Linear Algebra: Vectors, matrices, operations on matrices. Calculus: Particularly differentiation and integration. Probability and Statistics: Probability distributions, hypothesis testing, descriptive statistics, etc.
2	**Programming**	Python or R: Data science is commonly practiced using Python or R. Either language will suffice, but Python is more versatile and has a larger community. Libraries: Familiarize yourself with libraries like Pandas (for data manipulation), NumPy (for numerical computations), Matplotlib and Seaborn (for data visualization), and Scikit-learn (for machine learning).
3	**Data Manipulation**	Data Cleaning: Techniques for handling missing data, outliers, and inconsistencies. Data Wrangling: Extracting, transforming, and loading (ETL) data from various sources. Data Visualization: Presenting data effectively using plots, charts, and graphs.
4	**Domain Knowledge**	Depending on your interests, having some knowledge in the field you want to apply data science to (e.g., finance, healthcare, marketing) can be beneficial.
5	**Critical Thinkimg and Problem Solving**	Data science often involves comple problems the require critical thinking and creative soloutions. Pratice problem-solving skills and logical reasoning

2. Competencies

S/L	Competency	KSA Description
1.	**Data Science**	**Knowledge:** • Understanding of data, Types of data. • Knowledge of Structured, Semi-Structured and Un Structured Data **Skills:** • Ability to Analysis the data in Real Time. • Proficiency in utilizing data for Real time Application. **Attitudes:** • Appreciation for the importance of data in digital system.
2.	**Data Collection and Management**	**Knowledge:** • Understanding of Data Analytics and Visualization. **Skills:** • Designing a method for preprocessing the Data by handling Missing Values. **Attitudes;** • Appreciation for the role of Data Analytics and Visualizationin digital systems.
3.	**Data Analysis**	**Knowledge:** • Understanding of Statistics, Distribution, Machine Learning Algorithms. **Skills:** • Analyzing the data using Statistical Tool and Optimizing the behavior of data using Regression. **Attitudes:** • Valuing the importance of Real Time Data in digital system.
4.	**Model Selection**	**Knowledge:** • Understanding the Model selection, Validation. • Knowledge of Regression and Data Reduction.

		Skills: • Applying Regression and Reduction for Data Analytics. • Describing Feature Extraction, Cross Validation and behavioral models. **Attitudes:** • Openness to learning and using Feature Extraction, Data Reduction, Regression.
5.	**Supervised Learning**	**Knowledge:** • Understanding of KNN, PCA, Clustering. **Skills:** • Implementing various Classification Algorithms using Tools. **Attitudes:** • Recognizing the significance of Classification and Clustering Algorithms.

3. Syllabus

DATA SCIENCE
SEMESTER – II

Course Code	**M23MCA205D**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives:

- Understand data science and its applications.
- Understand the strategies of data collection and pre-processing.
- Apply statistics methods to develop models.
- Learn the evaluation metrics and techniques.

Module -1	
Introduction to Data Science- Introduction- Definition – History of Data Science -Understanding data: Introduction – Types of Data: Numeric – Categorical – Graphical – High Dimensional Data – Classification of digital Data: Structured, Semi-Structured and Un Structured - Example Applications. Sources of Data: Time Series – Transactional Data – Biological Data – Spatial Data – Social Network Data – Data Evolution.	L1, L2, L3
Module -2	
Data collection and Management- Introduction- Sources of data- Data collection and APIs- Exploring and fixing data- Data storage and management- using multiple data sources-Data Science in various fields - Examples - Impact of Data Science - Data Analytics Life Cycle - Data Science Toolkit -Applications of Data Science, Technologies for Data visualization.	L1, L2, L3
Module -3	
Data analysis- Introduction- Terminology and concepts-Introduction to statistics- Central tendencies and distributions- Variance- Distribution properties and arithmetic- Samples/CLT- Basic machine learning algorithms- Linear regression- SVM- Naive Bayes.	L1, L2, L3
Module -4	
Introduction to Model selection- Regularization- bias/variance tradeoff e.g. parsimony- AIC, BIC, Cross validation- Ridge regressions and penalized regression e.g. LASSO Data transformations- Dimension reduction- Feature extraction- Smoothing and aggregating.	L1, L2, L3
Module -5	
Supervised Learning- Regression- linear models- Regression trees- Time-series Analysis- Forecasting- Classification- classification trees- Logistic regression- separating hyper planes- k-NN Unsupervised Learning- Principal Components Analysis (PCA)- k-means clustering- Hierarchical clustering- Ensemble methods.	L1, L2, L3

Text Books

1. Cathy O Neil, RachelSchutt,2014,“DoingDataScience-StraightTalkfromthe Frontline”, Orielly

2. Jure Leskovek, Anand Rajaraman, Jeffrey Ullman, 2014 Mining of Massive Data Sets, Cambridge University Press

Reference Books

1. KevinMurphy,2013,Machinelearning: A Probabilistic Perspective

2. PeterBruce,AndreBruce,PracticalStatisticsforDataScientists,OriellySeries

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2 Introduction to Data Science**	• Introduction to Data Science • Data, Types of data, Structured, Semi-Structured and Un Structured Data • Ability to Analysis the data in Real Time. • Proficiency in utilizing data for Real time Application.
2	**Week 3-4-5 Data Collection and Management**	• Data Collection and Management • Data Analytics and Visualization. • Designing a method for preprocessing the Data by handling Missing Values.
3	**Week 6-7: Data Analysis**	• Data Analysis • Statistics, Distribution, Machine Learning Algorithms. • Analyzing the data using Statistical Tool and Optimizing the behavior of data using Regression.
4	**Week 8-9: Model Selection**	• Model Selection • Model selection, Validation, Regression and Data Reduction. • Applying Regression and Reduction for Data Analytics. • Describing Feature Extraction, Cross Validation and behavioral models.
5	**Week 10-11-12: Supervised Learning**	• Supervised Learning • KNN, PCA, Clustering. • Implementing various Classification Algorithms using Tools .

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming**	Assign programming tasks to reinforce practical skills associated with

	Assignments	competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding Data Science Fundamentals	Define Data Science: Understand the scope, significance, and interdisciplinary nature of data science. Explain Data Science Workflow: Describe the typical workflow in a data science project, including data collection, cleaning, exploration, modeling, and interpretation. Differentiate Data Science Roles: Identify the various roles in data science (e.g., data analyst, data engineer, data scientist) and their responsibilities.
2	Data Handling and Manipulation	Data Collection: Understand different data sources (structured and unstructured) and methods for data acquisition. Data Cleaning: Learn techniques for cleaning and preprocessing data, handling missing values, and dealing with outliers. Data Transformation: Perform data transformation tasks such as normalization, scaling, and encoding categorical variables.
3	Introduction to Machine Learning	Machine Learning Concepts: Differentiate between supervised and unsupervised learning and understand their applications. Model Building: Implement basic machine learning models (e.g., linear regression, decision trees, k-means clustering) using relevant tools. Model Evaluation: Evaluate model performance using metrics such as accuracy, precision, recall, and F1-score.
4	Tools and Technologies	Programming Skills: Develop proficiency in programming languages commonly used in data science, such as Python or R. Data Manipulation Libraries: Use libraries like Pandas, NumPy, and SQL for data manipulation and querying. Visualization Tools: Utilize visualization libraries and tools such as Matplotlib, Seaborn, and Tableau for data visualization. Machine Learning Frameworks: Gain hands-on experience with machine learning frameworks like Scikit-learn, TensorFlow, or Keras.

5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

CO's	**Description**
M23MCA205D.1	Apply the Data Science Lifecycle for Data Management.
M23MCA205D.2	Analyze statistical techniques to visualize the data and evaluate.
M23MCA205D.3	Design classifier model to predict future trends and to implement clustering techniques on the datasets
M23MCA205D.4	Implement Linear model selection methods for real time applications using modern tools

CO-PO- Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
M23MCA205D.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205D.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA205D.3	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA205D.4	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**
M23MCA205D	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**5**	**10**			**15**
Module 3			**10**		**10**
Module 4				**10**	**10**
Module 5				**5**	**5**
Total	**15**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				20
Module 2	**10**	**20**			30
Module 3			**20**		20
Module 4				**20**	20
Module 5				**10**	10
Total	**30**	**20**	**20**	**30**	**100**

10. Future with this Subject

The future of data science is believed to witness some of the biggest innovations seen in the last decade, starting from the data explosion to the growth of the internet of things (IoT) and social media.

Experts predict that in the next decade, the rise of machines with lead to the growth in usage and utility of computer systems and mobile devices.

1. AI Integration: AI incorporation for advanced predictive modeling and decision-making.
2. Ethical Data Use: Focus on responsible and transparent data handling.
3. Automation and Efficiency: Increased automation streamlining data processing, freeing up for innovation.
4. Interdisciplinary Collaboration: Collaboration with diverse disciplines enriching data science projects.
5. Edge Computing: Utilization of edge computing for real-time analytics in IoT applications.
6. Predictive Analytics: Advancements enabling accurate anticipation of future trends and behaviors.
7. Data Privacy & Security: Stricter measures and innovative encryption techniques for data protection.

2nd Semester	PROFESSIONAL ELECTIVE II(PE) BIG DATA ANALYTICS	M23MCA206A

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Data Collection**	Learning how to gather large amounts of data from different sources.
2	**Data Storage**	Understanding where and how to store the data so it can be easily accessed and managed.
3	**Data Cleaning**	Knowing how to clean the data by removing errors and inconsistencies to ensure it's useful.
4	**Data Analysis**	Learning methods and tools to examine and understand the data to find patterns, trends, and insights.
5	**Data Visualization**	Knowing how to create charts, graphs, and other visual tools to present the data findings clearly.
6	**Statistical and Mathematical Skills**	Applying statistics and math to make sense of the data.
7	**Programming Skills**	Using programming languages like Python or R to work with data.
8	**Communication Skills**	Learning how to effectively share data findings with others, especially those who may not be data experts.

2. Competencies

S/L	Competency	KSA Description
1	**Data Cleaning and Preparation**	Knowledge • **Data Management**: Understanding how to collect, store, and organize large datasets. • **Statistical Analysis**: Knowing statistical methods and how to apply them to analyze data. **Skills** • **Data Cleaning**: Ability to clean and preprocess data to ensure it is accurate and usable. • **Data Analysis**: Ability to use analytical techniques to explore data, identify patterns, and draw conclusions. Attitude • **Curiosity**: Eagerness to explore data and ask questions to uncover hidden insights. • **Attention to Detail**: Being meticulous and thorough in data analysis to ensure accuracy.
2	**Statistical Analysis**	**Knowledge** • **Programming**: Knowledge of languages such as Python, R, SQL, and others used for data manipulation and analysis, basic statistical analysis, such as calculating mean, median, mode, standard deviation, and performing regression analysis. **Skills:** • **Problem-Solving**: Applying analytical thinking to solve complex problems using data **Attitude** • Knowing how to create charts, graphs, and other visual tools to present the data findings clearly

3	**Data Mining**	**Knowledge** • Machine Learning: Understanding algorithms and techniques for predictive analytics. • Big Data Technologies: Familiarity with tools like Hadoop, Spark, and NoSQL databases. **Skills** • Programming: Writing and debugging code to manipulate and analyze data. • Data Visualization: Creating charts, graphs, and other visuals to present data insights clearly and effectively. **Attitude** • Critical Thinking: Being able to assess information critically and make reasoned judgments. • Adaptability: Willingness to learn and adapt to new tools, technologies, and methods in the rapidly evolving field of big data.
4	**Data Visualization Techniques**	**Knowledge** • Creating clear and effective visual representations of data insights. • Using tools like Tableau, Power BI, and Matplotlib • Designing dashboards and interactive reports **Skills** • Applying critical thinking to solve complex data-related problems. • Developing data-driven solutions • Optimizing algorithms and workflows **Attitude** • A strong desire to explore data and discover new insights. • Asking questions and seeking deeper understanding • Staying updated with the latest trends and technologies
5	**Data Interpretation**	**Knowledge** • Being able to interpret the results of your analysis and draw meaningful insights from the data is a crucial competency in big data analytics **Skills** • Effectively conveying data insights to non-technical stakeholders. • Writing reports and presenting findings • Using storytelling techniques to make data compelling **Attitude** • Working effectively with others in interdisciplinary teams to achieve common goals. • Sharing knowledge and supporting team members • Communicating effectively and fostering a collaborative environment

3. Syllabus

BIG DATA ANALYTICS **SEMESTER – II**			
Course Code	**M23MCA206A**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives:

1. Understand the Big Data Platform and its Use cases.
2. Provide an overview of Apache Hadoop, HDFS Concepts and Interfacing with HDFS.
3. Understand Map Reduce Jobs and Provide hands on Hadoop Eco System.

Module 1	
Introduction to Big Data: What is big data? Is the "big" part or the "data" art more important? How is big data different? How is big data more of the same? Risks of big data -why you need to tame big data -the structure of big data- exploring big data, most big data doesn't matter- filtering big data effectively mixing big data with traditional data- the need for standards today's big data is not tomorrow's big data. What web data reveals, Web data in action? A cross-section of big data sources and the value they hold.	**L1, L2**
Module 2	
Data Analysis: Evolution of analytic scalability – The convergence of the analytic and data environments, massively parallel processing systems, Cloud computing, Grid computing, Map reduce, Enterprise analytic sand box, Enterprise analytic data sets. Analytic Tools and Methods – The evolution of analytic tools and methods. Analysis approaches – Framing the problem, Statistical significance versus business importance. Enabling Analytic innovation – traditional approaches hamper innovation,	**L1, L2**
Module 3	
MapReduce and Hadoop Distributed Filesystem: A Weather Dataset, Analyzing the Data with Unix Tools, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes, The Design of HDFS, HDFS Concepts, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives.	**L2, L3**
Module 4	
Introduction to Hadoop and its Operations: Administering hadoop- HDFS, Monitoring, and Maintenance. Pig- Installing and running pig, Comparison with Databases, pig latin, User-defined functions. Hive- Installing Hive, running hive, Comparison with traditional databases, HiveQL, Querying data.	**L3, L4**
Module 5	
Recommendation Systems and Mining Social- Network Graphs: A model for recommendation systems, Content- based recommendation, Collaborative filtering, Dimensionality Reduction, The Netflix challenge. Mining social-network graphs- Social networks as graphs, Clustering of social-network graphs, Partitioning of graphs, Neighborhood properties of graphs.	**L3, L4**

Text Books:

1. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced
Analytics”, John Wiley & sons, 2014.
2. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets” 2nd edition, Cambridge University Press, 2016.
3. Tom White, “Hadoop: The Definitive Guide”, O’reily Media, 4th edition, 2015.

Reference Books:

1. Paul Zikopoulos, Chris Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming
Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw Hill Professional, 2012.
2. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, Pete Warden, Big Data Glossary, O“Reilly,2016.
3. Chuck Lam, “Hadoop in Action”, Dream tech Press, 2nd edition 2014.

Journals/Magazines:

1. R. Almutiri, S. Alhabeeb, S. Alhumud and R. U. Khan, "A survey of machine learning for big data processing," Journal on Big Data, vol. 4, no.2, pp. 97–111, 2022.
2. Zhong W, Yu N, Ai C. Applying Big Data Based Deep Learning System to Intrusion Detection. Big Data Mining and Analytics, 2020.

3. https://industrywired.com/top-10-big-data-and-artificial-intelligence-magazines-and-publications/
4. https://www.admin-magazine.com/tags/view/Hadoop

Web/Digital resources:

1. https://www.tutorialspoint.com/big_data_tutorials.html
2. https://www.linkedin.com/learning/topics/big-data?trk=lynda_redirect_learning
3. https://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.html
4. https://bigdatauniversity.com

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2 Getting Started with BDA**	• Exploring big data, most big data doesn't matter- filtering big data effectively mixing big data with traditional data • students will know the process • Problem Solving, Documentation
2	**Week 3-4 Working with Data**	• Knowledge of Data Modeling, • Create, Read, Update and Delete. • Enterprise analytic data sets. Analytic Tools and Methods
3	**Week 5-6 MapReduce and Hadoop**	• Knowledge of MapReduce and Hadoop • Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes, The Design of HDFS, HDFS Concepts, • improves Critical Thinking, Innovation
4	**Week 7-8 Hadoop Operations**	• Knowledge of what if analysis of data processed • Pig- Installing and running pig, Comparison with Databases, pig latin, User-defined functions. Hive- Installing Hive, running hive, • Analytical Thinking
6	**Week 9-10 Recommendation system and social networks**	• A model for recommendation systems • A model for recommendation systems, Content- based recommendation, Collaborative filtering, Dimensionality Reduction • Social networks as graphs, clustering of social-network graphs, Partitioning of graphs, Neighborhood properties of graphs.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Structured Curriculum**	Develop a well-structured curriculum that covers fundamental AI concepts, including machine learning, neural networks, natural language processing, and computer vision. Outline learning objectives, topics, and milestones to guide students through the course.
2	**Active Learning Techniques**	Incorporate active learning techniques such as problem-based learning, case studies, and group discussions to engage students and encourage participation. Provide opportunities for students to apply AI concepts to real-world problems through projects and assignments.
3	**Hands-on Projects**	Offer hands-on projects where students can experiment with AI algorithms, tools, and datasets. Provide access to relevant software and resources, such as Python programming environments, AI libraries, and cloud computing platforms.
4	**Guest Lectures and Industry Connections**	Invite guest speakers from industry, academia, and research institutions to share their expertise and experiences in AI. Organize field trips, industry visits, or virtual seminars to expose students to real-world AI applications and career opportunities.
5	**Interactive Lectures and Demonstrations**	Use a variety of teaching methods, including interactive lectures, demonstrations, and multimedia presentations, to explain complex AI concepts. Use visual aids, simulations, and interactive tutorials to illustrate key concepts and algorithms.

6	**Student-Centered Learning**	Empower students to take ownership of their learning by encouraging independent inquiry, research, and exploration. Provide opportunities for self-directed learning through online resources, tutorials, and project-based learning platforms.
7	**Assessment and Feedback**	Implement a variety of assessment methods, including quizzes, exams, projects, and presentations, to evaluate students' understanding of AI concepts and their ability to apply them. Provide constructive feedback to help students improve their skills and knowledge.
8	**Ethical and Social Implications**	Integrate discussions on the ethical and social implications of AI into the curriculum. Encourage students to critically evaluate the impact of AI on society, privacy, bias, fairness, and employment.
9	**Peer Learning and Collaboration**	Foster a collaborative learning environment where students can work together in teams, share ideas, and collaborate on projects. Encourage peer-to-peer learning, mentorship, and peer review to promote knowledge sharing and teamwork.
10	**Continuous Improvement and Updates**	Continuously update the course content and teaching materials to reflect the latest advancements in AI research, technologies, and applications. Seek feedback from students and colleagues to identify areas for improvement and innovation.

6. **Assessment Details (both CIE and SEE)**

CIE Split up for Professional Elective Course (PE)

	Components	**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	**Learning Objectives**	**Description**
1	**Big Data – Introduction**	**Understand Big Data**: Learn what big data is, where it comes from, and why it's important. **Collect Data**: Know how to gather large amounts of data from various sources. **Store Data**: Learn how to save data in a way that it can be easily accessed and managed. **Clean Data**: Be able to fix errors and organize data so it can be used effectively. **Analyse Data:** Use tools and methods to look at data and find useful information. **Visualize Data:** Create charts and graphs to show data findings clearly. **Use Statistical Methods:** Apply basic statistics to understand data better

APPROVED · Dean Academics ·

Dean Academics
MIT Mysore

Principal
MIT Mysore

2	**Data Analysis**	**Data:** Examine large sets of data to see what information is there. **Identify Patterns:** Spot trends and patterns that tell you something important. **Draw Conclusions:** Make sense of the data to understand what it means and how it can be used to make decisions. **Use Tools and Methods:** Apply different techniques and tools to analyze the data effectively. **Solve Problems:** Use the insights gained from data analysis to address real-world problems
3	**MapReduce and Hadoop File System and operations**	**Store Large Datasets:** Understand how HDFS stores data by breaking it into smaller pieces and distributing them across multiple computers. **Access and Manage Data:** Learn how to retrieve and manage data stored in HDFS efficiently. **Ensure Data Reliability:** Understand how HDFS keeps multiple copies of data to protect against hardware failures and ensure data is always available.
4	**Social networks**	**Ethical Mind set:** Commitment to using data responsibly and maintaining high ethical standards. Respecting privacy and confidentiality. Promoting transparency and accountability **Collaboration:** Working effectively with others in interdisciplinary teams to achieve common goals. Sharing knowledge and supporting team members Communicating effectively and fostering a collaborative environment

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA206A.1	Comprehend the significance, structure and sources of Big data.
M23MCA206A.2	Explore avenues for analytical scalability using analytical tools and methods.
M23MCA206A.3	Analyze and Design data with Hadoop tools and different operations on Hadoop
M23MCA206A.4	Apply social networking using map reduction technique using modern techniques

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA206A.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206A.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206A.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206A.4	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206A	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**5**	**10**			**15**
Module 3			**10**		**10**
Module 4			**5**		**5**
Module 5				**10**	**10**
	15	**10**	**15**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**

Module 2	**10**	**20**			**30**
Module 3			**20**		**20**
Module 4			**10**		**10**
Module 5				**20**	**20**
	30	**20**	**30**	**20**	**100**

10. Future with this Subject

- **Increased Demand for Skills**
  - **High Demand**: As more companies rely on data for decision-making, the need for skilled big data analysts will keep growing.
  - **Career Opportunities**: There will be more job opportunities in various industries like healthcare, finance, retail, and tech.
- **Advanced Technologies**
  - **AI and Machine Learning**: Courses will include more about artificial intelligence and machine learning, as these technologies are becoming essential for analyzing big data.
  - **Real-Time Analytics**: Learning to analyze data in real-time will become more important as businesses need immediate insights.
- **Practical Applications**
  - **Hands-On Experience**: Courses will offer more practical, hands-on experience with real-world data projects to prepare students for the job market.
  - **Industry Collaboration**: Increased partnerships with businesses to ensure that the curriculum meets current industry needs.
  - **Data Ethics**: A stronger focus on the ethical use of data, privacy concerns, and data security will be included as these issues become more critical.
- **Broader Access and Flexibility**
  - Online Learning: More courses will be available online, making it easier for people around the world to learn big data analytics.
  - Flexible Learning: Courses will offer flexible learning options, allowing students to learn at their own pace.
- **Interdisciplinary Approach**
  - **Integration with Other Fields**: Big data analytics will be combined with other fields such as business, engineering, and social sciences to provide a more comprehensive education.

2nd Semester	PROFESSIONAL ELECTIVE II(PE) CYBER SECURITY	M23MCA206B

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic knowledge**	Basic knowledge of computer systems, networks, and the internet.
2.	**security concepts**	Familiarity with security concepts like confidentiality, integrity, and availability.
3.	**vulnerabilities in systems**	Ability to analyze and identify vulnerabilities in systems and networks.
4.	**Basic knowledge of digital forensics principles.**	Conducting digital forensics investigations, including evidence collection and analysis.
5.	**Basic understanding of mobile and wireless technologies**	Analyzing and responding to security incidents involving mobile devices.
6.	**Awareness of web threats and security**	Addressing security and privacy implications for organizations in the context of social computing and web threats.
7.	**Understanding of basic data privacy concepts and principles.**	Ability to analyze and assess privacy risks and threats in different domains.
8.	**Awareness of common data privacy attacks**	Competence in addressing privacy issues and challenges in various domains such as medical and financial sectors.

2. Competencies

S/L	Competency	KSA Description
1.	**Basic Cyber Security Concepts**	**Knowledge:** • Understanding of the layers of security. • Awareness of vulnerabilities, threats, and harmful acts in cyberspace. **Skills:** • Ability to identify and assess vulnerabilities and threats. • Capability to analyze and prioritize security measures based on risks. **Attitudes:** • Awareness of the importance of cyber security for personal and organizational safety. • Commitment to staying updated on emerging cyber threats and security best practices.
2.	**Cyberspace Law & Cyber Forensics**	**Knowledge:** • Understanding of cyber security regulations and international law • Knowledge of the Indian cyberspace landscape and the National Cyber Security Policy **Skills:** • Ability to interpret and comply with cyber security regulations and legal requirements. • Proficiency in conducting digital forensics investigations. **Attitudes:** • Respect for legal and ethical standards in cyber investigations. • Patience and attention to detail required for thorough forensic analysis
3.	**Cybercrime, Mobile and Wireless**	**Knowledge:** • Understanding of the proliferation and trends of mobile and wireless devices. • Knowledge of common cybercrimes involving mobile and wireless

	Devices	computing. **Skills:** • Ability to implement security measures to protect mobile devices and data • Proficiency in detecting and mitigating attacks targeting mobile devices. **Attitudes:** • Vigilance regarding the security risks associated with mobile technologies. • Pro-activeness in adopting security policies and measures to safeguard mobile assets
4.	**Cyber Security Organizational Implications:**	**Knowledge:** • Understanding of the costs and implications of cybercrimes for organizations. • Knowledge of web threats, security, and privacy implications for organizations **Skills:** • Ability to assess and mitigate organizational cyber risks. • Competence in developing and implementing security policies and measures. **Attitudes:** • Commitment to protecting organizational assets and data privacy. • Openness to collaboration and knowledge sharing within the organization and across industry sectors
5.	**Privacy Issues**	**Knowledge:** • Understanding of fundamental data privacy concepts and principles. • Awareness of privacy policies, specifications, and regulations in different domains. **Skills:** • Ability to assess privacy risks and develop strategies to protect sensitive data. • Competence in drafting and implementing privacy policies and compliance measures. **Attitudes:** • Respect for individual privacy rights and obligations to protect personal data. • Sensitivity to the ethical implications of data handling and privacy breaches.

3. Syllabus

CYBER SECURITY **SEMESTER – II**			
Course Code	**M23MCA206B**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Course objectives: This course will enable students to: ➢ Understand various types of cyber-attacks and cyber-crimes ➢ Learn threats and risks within context of the cyber security ➢ Have an overview of the cyber laws & concepts of cyber forensics ➢ Study the defensive techniques against these attacks			
Module -1			
Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.			L1, L2, L3
Module -2			

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy. Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics	L1, L2, L3
Module -3	
Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.	**L1, L2, L3**
Module -4	
Cyber Security: Organizational Implications: Introduction cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations.	L1, L2, L3
Module -5	
Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc.	L1, L2, L3

Text Books:

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley

2. B.B.Gupta, D.P.Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335, 2018.

REFERENCES:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&F Group

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction to Cyber Security**	Basic Cyber Security Concepts: Layers of security, Vulnerability, Threat, Harmful acts, Internet Governance – Challenges and Constraints. Computer Criminals, CIA Triad, Motive of attackers, Active attacks, Passive attacks, Software attacks, Hardware attacks,
2	**Week 3-4: Cyber Security, Cyberspace and the Law**	Cyber Threats-Cyber Warfare, Cyber Crime, Cyber Terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy. Cyberspace and the Law & Cyber Forensics: Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy. Historical background of Cyber forensics, Digital Forensics Science.
3	**Week 5-6: Cyber Forensics and Cybercrime:**	The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics. Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era,
4	**Week 7-8: Cybercrime:**	Cybercrime: Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

5	**Week 9-10: Cyber Security:**	Cyber Security: Organizational Implications: Introduction cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, Social media marketing: Security risks and perils for organizations, social computing and the associated challenges for organizations.
6	**Week 11-12: Privacy Issues:**	Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of the concepts.
3	**Collaborative Learning**	It provides culture of information sharing, cooperation, and continuous improvement, ultimately strengthening the collective cyber security posture
4	**Higher Order Thinking (HOTS) Questions:**	It is not only challenging individuals to think critically about cyber security issues but also encourage them to apply their knowledge in real-world scenarios, fostering a deeper understanding of the field.
5	**Problem-Based Learning (PBL)**	PBL prepares students to tackle the dynamic and evolving challenges of cyber security effectively. It fosters a deep understanding of cyber security principles, encourages lifelong learning, and cultivates the skills needed to thrive in the cyber security profession.
6	**Multiple Representations**	Cyber security professionals can gain deeper insights into security-related data, communicate complex concepts more effectively, and make more informed decisions to protect against cyber threats.
7	**Real-World Application**	These real-world applications demonstrate the diverse ways in which cyber security principles and practices are applied to safeguard digital assets, mitigate risks, and defend against evolving cyber threats in today's interconnected world.
8	**Flipped Class Technique**	It promotes active engagement, self-directed learning, critical thinking, and practical skills development, preparing students for the dynamic and evolving challenges of the cyber security profession.
9	**Programming Assignments**	It provides students with practical skills and real-world experience in applying programming concepts to address security challenges.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2(TWO) test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.

2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Basic Cyber Security Concepts**:	Understand the layers of security and their significance in protecting systems and data. Define vulnerability, threat, and harmful acts in the context of cyber security.
2	**Cyberspace Law & Cyber Forensics**:	Understand the Indian cyberspace and the National Cyber Security Policy. Trace the historical background of cyber forensics and digital forensics science.
3	**Cybercrime: Mobile and Wireless Devices**:	Identify common cybercrimes involving mobile and wireless computing, such as credit card frauds. Recognize security challenges posed by mobile devices and understand authentication service security.
4	**Cyber Security: Organizational Implications**	Evaluate the cost of cybercrimes and intellectual property rights (IPR) issues for organizations. Identify web threats and their implications for organizational security and privacy.
5	**Privacy Issues**	Recognize data privacy attacks and the risks associated with data linking and profiling. Evaluate privacy policies and their specifications, including privacy policy languages.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA206B.1	Analyze cyber-attacks, types of cybercrimes, cyber laws and also how to protect them self and ultimately the entire Internet community from such attacks.
M23MCA206B.2	Interpret and forensically investigate security incidents
M23MCA206B.3	Apply policies and procedures to manage Privacy issues
M23MCA206B.4	Design and develop secure software modules

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA206B.1	-	3	-	-	-	-	-	-
M23MCA206B.2	-	3	-	-	-	-	-	-
M23MCA206B.3	3	-	-	-	-	-	-	-
M23MCA206B.4	-	-	3	-	-	-	-	-
M23MCA206B	3	3	3	-	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**

Module 1	**10**				**10**
Module 2	**5**				**5**
Module 3		**10**			**10**
Module 4		**5**	**10**		**15**
Module 5				**10**	**10**
	15	**15**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2	**10**				**10**
Module 3		**20**			**20**
Module 4		**10**	**20**		**30**
Module 5				**20**	**20**
	30	**30**	**20**	**20**	**100**

Conditions for SEE Paper Setting:
Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject.

- **Advanced Threats**: Understand the evolving landscape of cyber threats, including advanced persistent threats (APTs), ransomware, and supply chain attacks.
- **Cyber Defense**: Develop skills in implementing robust cyber security measures, including network security, endpoint protection, encryption, and access control.
- **Incident Response**: Acquire proficiency in incident detection, response, and recovery techniques to mitigate the impact of cyber attacks effectively.
- **Continuous Learning**: Cultivate a mindset of continuous learning and adaptation to keep pace with evolving cyber threats and defensive techniques.
- **Collaboration**: Embrace collaboration with peers, industry experts, and relevant authorities to share knowledge and best practices in cyber security.
- **Cyber security Analyst**: Pursue a career as a cyber security analyst, responsible for monitoring, analyzing,
- **Ethical Hacker/Penetration Tester**: Explore opportunities as an ethical hacker or penetration tester, helping organizations identify and remediate vulnerabilities in their systems.
- **Security Consultant**: Become a security consultant, advising organizations on cyber security strategies, risk management, and compliance with regulatory requirements.
- **Research and Development**: Engage in research and development in cyber security, contributing to the advancement of security technologies and techniques.
- **Cyber security Automation**: Expect increased adoption of automation and AI-driven solutions for threat detection, response, and security operations.
- **Zero Trust Architecture**: Witness the proliferation of zero trust architecture, where access to resources is strictly controlled and verified, regardless of location or user identity.

2nd **Semester**	**PROFESSIONAL ELECTIVE II(PE)** **ARTIFICIAL INTELLIGENCE**	**M23MCA206C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic Programming Skills**	While not always mandatory, having a foundational understanding of programming concepts can be beneficial. Python is widely used in AI due to its simplicity and robust libraries for machine learning and data manipulation.
2.	**Mathematics Fundamentals**	Familiarity with basic mathematics concepts such as algebra, calculus, probability, and statistics is essential for understanding the algorithms and models used in AI.
3.	**Understanding of Data**	Knowledge of how data is collected, structured, and processed is crucial in AI. This includes familiarity with databases, data formats, and data preprocessing techniques.
4.	**Curiosity and Critical Thinking**	AI involves problem-solving and continuous learning. Having a curious mindset and the ability to think critically about different AI applications and their implications is important.
5.	**Books and Research Papers**	Reading introductory books and research papers on AI can provide a deeper understanding of fundamental concepts and current trends in the field.
6.	**Hands-on Projects**	Building AI projects, even simple ones, is invaluable for gaining practical experience and reinforcing theoretical concepts. There are many resources and datasets available online for practicing AI projects.
7.	**Community Engagement**	Joining AI communities, forums, or local meetups can provide opportunities to learn from others, ask questions, and stay updated on the latest developments in the field.

2. Competencies

S/L	Competency	KSA Description
1.	**Basic AI Concepts, Programming Curiosity**	**Knowledge:** 30. Understanding fundamental concepts such as machine learning, neural networks, deep learning, natural language processing, and computer vision. **Skills:** 31. Proficiency in a programming language commonly used in AI development, such as Python, along with basic programming concepts and syntax. **Attitudes:** 32. A curious mindset to explore and learn about new AI concepts, techniques, and applications.
2.	**AI Applications, Data Handling**	**Knowledge:** 33. Knowledge of various real-world applications of AI across industries such as healthcare, finance, marketing, autonomous vehicles, etc. **Skills:** 34. Ability to manipulate and analyze data using libraries like pandas, NumPy, and scikit-learn, including tasks like data cleaning, feature extraction, and visualization. **Attitudes:** • Willingness to learn from failures and mistakes, and continuously improve skills and knowledge in AI.

3.	**Ethical Considerations, Machine Learning Basic**	**Knowledge:** ● Awareness of ethical issues in AI, including bias, fairness, transparency, privacy, and accountability. **Skills:** ● Basic knowledge of machine learning algorithms, including supervised learning, unsupervised learning, and evaluation metrics. **Attitudes:** ● Consciousness of ethical implications in AI development and deployment, and a commitment to responsible and ethical AI practices.
4.	**AI Tools and Technologies, Problem-Solving**	**Knowledge:** ● Familiarity with popular AI development tools, libraries, and frameworks like TensorFlow, PyTorch, scikit-learn, etc. **Skills:** ● Skill in formulating AI problems, selecting appropriate algorithms, and implementing solutions to address specific tasks or challenges. **Attitudes:** ● Flexibility to adapt to changes and advancements in AI technologies and methodologies.
5.	**Data Fundamentals, Critical Thinking**	**Knowledge:** ● Understanding of data types, structures, and preprocessing techniques relevant to AI, including data cleaning, transformation, and feature engineering. **Skills:** ● Ability to critically evaluate AI solutions, identify potential biases or limitations, and propose improvements or alternatives. **Attitudes:** ● Readiness to collaborate with others, share knowledge, and work in interdisciplinary teams to solve AI-related problems.

3. Syllabus

ARTIFICIAL INTELLIGENCE **SEMESTER – II**			
Course Code	**M23MCA206C**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Course objectives: ● Gain a historical perspective of AI and its foundations. ● Become familiar with basic principles of AI toward problem solving ● Get to know approaches of inference, perception, knowledge representation, and learning			
Module -1			
Introduction: Artificial Intelligence, The Foundations of Artificial Intelligence, History of Artificial Intelligence, The State of the Art in AI. **Intelligent Agent**: Agents and Environments, Good Behavior: Concept of Rationality, The Nature of Environments, The Structure of Agents. **Problem Solving:** Problem-Solving Agents, Example Problems.			L1, L2, L3
Module -2			

Problem Solving: Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions. **Logical Agents**: Knowledge–Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving, Effective Propositional Model Checking, Agent Based on Propositional Logic.	L1, L2, L3
Module -3	
First Order Logic: Representation Revisited, Syntax and Semantics of First Order Logic, Using First Order Logic, Knowledge Engineering in First Order Logic. **Inference in First Oder**: Propositional vs. First Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.	L1, L2, L3
Module -4	
Uncertain Knowledge and Reasoning: **Quantifying Uncertainty**: Acting under Uncertainty, Basic Probability Notation, Inference using Full Joint Distributions, Independence, Baye's Rule and Its Use, Wumpus World Revisited. **Learning from Examples:** Forms of Learning, Supervised Learning, Learning Decision Trees, Regression and Classification with Linear Models, Artificial Neural Networks, Support Vector Machines.	L1, L2, L3
Module -5	
Natural Language Processing: Language Model, Text classification, Information Retrieval and Extraction. Case Study: NLP Techniques **Perception**: Image Formation, Early Image Processing Operation, Object Recognition by Appearance, Reconstructing the 3D World. Case Study: Image Processing In Agriculture **Robotics:** Introduction, Robot Hardware, Robotic Perception, Robotic Software Architecture, Application Domain**.** Case Study: Robotic Cars	L1, L2, L3

Text Books:

1. Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Reference Books:

1. Elaine Rich, Kevin Knight, Artificial Intelligence, 3rd edition, Tata McGraw Hill,2013
2. R. B Mishra, Artificial intelligence PHI Learning Pvt. Ltd., 2010

Web links and Video Lectures (e-Resources)

1. https://nptel.ac.in/courses/106/105/106105077/
2. https://archive.nptel.ac.in/courses/106/105/106105152/
3. https://archive.nptel.ac.in/courses/106/105/106105158/
4. https://archive.nptel.ac.in/courses/117/105/117105135/
5. https://archive.nptel.ac.in/courses/107/106/107106090/

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Introduction, Intelligent Agents, Problem-solving - I**	35. Foundations of Artificial Intelligence 36. Provides a foundational understanding of AI concepts and its evolution 37. Algorithm design and programming, essential for developing intelligent systems
2	**Week 3-4: Problem Solving - II**	38. Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) search strategies, Heuristic functions 39. Provides knowledge about the principles and algorithms underlying these strategies, enabling effective application in AI systems. 40. Problem-solving through searching for solutions using both uninformed and informed search strategies, and developing heuristic functions to guide efficient search processes.

3	**Week 5-6: Logical Agent, First Order Logic**	41. Knowledge–based agents, : Representation Revisited, Syntax and Semantics of first order logic 42. Formal logic principles, predicate calculus, and methods for representing and manipulating knowledge, enabling the design of more robust and interpretable systems. 43. creating intelligent systems capable of logical reasoning and decision-making
4	**Week 7-8: Inference in First Order Logic, Quantifying Uncertainty**	44. Propositional vs. First order inference, Unification and Lifting, Basic 45. Probability Notation, Inference using Full Joint Distributions 46. Drawing logical conclusions from a set of premises, including resolution, unification, and theorem proving techniques 47. Reasoning and problem-solving, capable of intelligent decision-making and knowledge representation.
5	**Week 9-10: Application of AI: Natural Language Processing, Computer Vision, Robotics.**	48. Building AI application on 49. data preprocessing, feature engineering, model selection, evaluation metrics, and deployment strategies 50. identifying suitable AI techniques and algorithms for solving real-world problems across various domains such as healthcare, finance, marketing, and robotics

5. **Teaching-Learning Process Strategies**

S/L	TLP Strategies:	Description
1	**Structured Curriculum**	Develop a well-structured curriculum that covers fundamental AI concepts, including machine learning, neural networks, natural language processing, and computer vision. Outline learning objectives, topics, and milestones to guide students through the course.
2	**Active Learning Techniques**	Incorporate active learning techniques such as problem-based learning, case studies, and group discussions to engage students and encourage participation. Provide opportunities for students to apply AI concepts to real-world problems through projects and assignments.
3	**Hands-on Projects**	Offer hands-on projects where students can experiment with AI algorithms, tools, and datasets. Provide access to relevant software and resources, such as Python programming environments, AI libraries, and cloud computing platforms.
4	**Guest Lectures and Industry Connections**	Invite guest speakers from industry, academia, and research institutions to share their expertise and experiences in AI. Organize field trips, industry visits, or virtual seminars to expose students to real-world AI applications and career opportunities.
5	**Interactive Lectures and Demonstrations**	Use a variety of teaching methods, including interactive lectures, demonstrations, and multimedia presentations, to explain complex AI concepts. Use visual aids, simulations, and interactive tutorials to illustrate key concepts and algorithms.
6	**Student-Centered Learning**	Empower students to take ownership of their learning by encouraging independent inquiry, research, and exploration. Provide opportunities for self-directed learning through online resources, tutorials, and project-based learning platforms.
7	**Assessment and Feedback**	Implement a variety of assessment methods, including quizzes, exams, projects, and presentations, to evaluate students' understanding of AI concepts and their ability to apply them. Provide constructive feedback to help students improve their skills and knowledge.
8	**Ethical and Social Implications**	Integrate discussions on the ethical and social implications of AI into the curriculum. Encourage students to critically evaluate the impact of AI on society, privacy, bias, fairness, and employment.

9	**Peer Learning and Collaboration**	Foster a collaborative learning environment where students can work together in teams, share ideas, and collaborate on projects. Encourage peer-to-peer learning, mentorship, and peer review to promote knowledge sharing and teamwork.
10	**Continuous Improvement and Updates**	Continuously update the course content and teaching materials to reflect the latest advancements in AI research, technologies, and applications. Seek feedback from students and colleagues to identify areas for improvement and innovation.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**20**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	**Learning Objectives**	**Description**
1	**Understand the foundational concepts**	Understand the foundational concepts and historical development of artificial intelligence, including the principles of intelligent agents, problem-solving strategies, and the evolution of AI technologies.
2	**Develop proficiency in knowledge representation and reasoning techniques**	Develop proficiency in knowledge representation and reasoning techniques, including predicate logic, onto logies, and common-sense reasoning, to effectively model and solve complex problems in AI applications.
3	**Gain practical knowledge and skills in machine learning fundamentals**	Gain practical knowledge and skills in machine learning fundamentals, including supervised and unsupervised learning algorithms, evaluation metrics, and techniques for mitigating issues such as overfitting and regularization.
4	**Explore the principles and architectures of neural networks and deep learning models**	Explore the principles and architectures of neural networks and deep learning models, including perceptron's, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), and understand their applications in various domains such as computer vision and natural language processing

5	**Apply Python programming skills to implement AI algorithms and frameworks**	Apply Python programming skills to implement AI algorithms and frameworks introduced throughout the course, enabling students to develop hands-on experience in building AI systems and applications
6	**Analyze and discuss real-world applications of artificial intelligence across diverse domains**	Analyze and discuss real-world applications of artificial intelligence across diverse domains, including robotics, healthcare, and ethical considerations, to understand the societal impact and ethical implications of AI technologies.

8. Course Outcomes (COs) and Mapping with PO

Course Outcomes (COs)

COs	**Description**
M23MCA206C.1	Understand and Apply knowledge of AI fundamentals and Intelligent agent types.
M23MCA206C.2	Analyze and apply the use of logic and knowledge representation for problem solving.
M23MCA206C.3	Formulate knowledge reasoning using propositional logic and first order logic
M23MCA206C.4	Analyze Quantifying uncertainty using probability notations.

CO-PO Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
M23MCA206C.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206C.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206C.3	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA206C.4	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206C	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2	**5**	**10**			**15**
Module 3		**5**			**5**
Module 4			**10**		**10**
Module 5				**10**	**10**
	15	**15**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2	**10**	**20**			**30**
Module 3		**10**			**10**
Module 4			**20**		**20**
Module 5				**20**	**20**
	30	**30**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

- Innovation and Advancements: As students become introduced to AI concepts at an early stage, they'll contribute to a culture of innovation and drive advancements in AI technology. This can lead to breakthroughs in areas such as healthcare, transportation, finance, and more, solving complex problems and improving quality of life.
- Workforce Readiness: Introducing AI in education ensures that students are equipped with the knowledge and skills needed to thrive in a future where AI is ubiquitous. This prepares them for AI-related jobs across various sectors, ranging from data science and machine learning engineering to AI ethics and policy-making.
- Ethical AI Development: Education on AI ethics and responsible AI practices cultivates a generation of professionals who prioritize ethical considerations in AI development and deployment. This includes addressing biases, ensuring transparency and accountability, and promoting fairness and inclusivity in AI systems.
- Cross-disciplinary Collaboration: Introduction to AI fosters collaboration across different disciplines, as AI intersects with fields such as computer science, mathematics, engineering, psychology, sociology, and more. Collaborative efforts lead to innovative solutions that tackle complex challenges from multiple perspectives.
- Entrepreneurship and Startups: Students introduced to AI may be inspired to pursue entrepreneurship and create AI-driven startups, addressing niche markets or disrupting existing industries. This entrepreneurial spirit contributes to economic growth, job creation, and technological innovation.
- AI Education Accessibility: Advancements in AI education technologies, such as online courses, interactive tutorials, and AI-driven personalized learning platforms, make AI education more accessible to learners worldwide. This democratization of AI education empowers individuals from diverse backgrounds to acquire AI skills and knowledge.
- Global Impact: Introduction to AI transcends geographical boundaries, empowering learners from different regions to contribute to global AI initiatives. Collaboration among international institutions, researchers, and students accelerates AI research, innovation, and knowledge-sharing on a global scale.
- AI for Social Good: Educating students on the potential of AI for social good encourages them to apply AI technologies to address pressing societal challenges, such as healthcare disparities, environmental sustainability, education accessibility, and poverty alleviation. AI-driven solutions have the potential to create positive social impact and promote inclusive development.
- Continuous Learning and Adaptation: In a rapidly evolving field like AI, continuous learning and adaptation are essential. Introduction to AI instills a culture of lifelong learning, encouraging individuals to stay updated with the latest advancements, trends, and best practices in AI throughout their careers.
- Ethical Leadership and Governance: As future leaders and policymakers, students introduced to AI play a crucial role in shaping ethical AI governance frameworks and policies.

2nd Semester	**PROFESSIONAL ELECTIVE II(PE) STATISTICAL AND NUMRERICAL METHODS**	**M23MCA206D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Combinatorics & Discrete Mathematics**	Basic knowledge of Combinatorics, probability theory and types of functions
2	**Linear algebra**	Familiarity with linear algebra and basic counting methods such as binomial coefficient is assumed
3	**Mathematics**	Proficiency in algebra for Boolean expressions implification using K-map techniques
4	**Fundamental Mathematics Knowledge**	Knowledge of basic algebraic mathematics like union intersections permutations and combinations and binomial Theorem.
5	**Relations and Functions**	Ability to analyze Cartesian product of set and identify the relations
6	**Algebra**	Proficiency in algebraic manipulations, factorization techniques, and solving algebraic equations is necessary for dealing with functions effectively.
7	**Matrices and Determinants**	While not directly related to functions, knowledge of matrices and determinants can be helpful in certain types of function problems.
8	**Probability and Statistics**	Understanding basic probability concepts and statistics can be useful in certain types of function problems that involve probability distributions or data analysis.
9	**Previous Coursework**	Completion of introductory courses in Mathematics or a related field

2. Competencies

S/L	Competency	KSA Description
1	**Random variable and probability distribution**	**Knowledge:** • Understanding the algorithm development, data analysis, machine learning, and simulation modeling. **Skills:** • Applying Probability to analyze data analysis, statistical inference, and machine learning **Attitudes:** • Valuing the importance in decision and estimation problems, and constructs computer algorithms for generating observations from the various distributions
2	**Design of experiments**	**Knowledge:** • Design of experiments, Block Design, Latin square design, Graeco Latin Squares **Skills:** • Using statistical theory of the design of experiments **Attitudes:** • Appreciation for the Latin and Graeco-Latin squares have an important application to the statistical theory of the design of experiments.
3	**Estimation**	**Knowledge:** • Statistics Inference, Estimation error-bias

		Skills: • Using point estimate definition is a calculation where a sample statistic is used to estimate or approximate an unknown population parameter **Attitudes:** • Appreciation for analyzing the interval estimation is the use of sample data to estimate an interval of possible values of a parameter of interest.
4	**Reliability Engineering**	**Knowledge:** • Understanding of Concepts of Reliability, Reliability of systems **Skills:** • Applying Reliability engineering can be applied to many business functions, from design to maintenance **Attitudes:** • Valuing the importance of reliability is a critical factor that focuses on the ability of a system, product, or process to perform its intended functions without malfunctioning or breaking down consistently.
5	**Stochastic Process**	**Knowledge:** • Markov chain and related problems. Queuing theory- Poisson queuing system, **Skills:** • Applying Stochastic Process to analyze Image Processing, Neuroscience, Bio Informatics, Financial Management, Statistics **Attitudes:** • Valuing the importance of Stochastic Processes in real-time mathematical model of systems which has a continuous random varying nature
6	**M/M/1 and M/M/s queuing models.**	**Knowledge:** • Little law. Discussion of M/M/1 and M/M/s queuing models. **Skills:** • Ability to apply Queuing Theory in model **Attitudes:** • Valuing the importance of M/M/1 Queue: The M/M/1 queue represents a single-server queuing system with Poisson arrivals, exponentially distributed service times, and a first-come-first-served discipline.

3. Syllabus

STATISTICAL AND NUMERICAL METHODS **SEMESTER – II**			
Course Code	**M23MCA206D**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Course objectives: • To familiarize the important tools of advanced numerical methods required to analyze the engineering problems. • Acquire the knowledge of probability and statistics applied in their core domain • To apply the knowledge of statistical techniques, stochastic process and queuing theory • to offer solutions the engineering problems • Improve their Mathematical Thinking and acquire skills required for sustained lifelong learning.			
Module -1			

Dean Academics MIT Mysore

APPROVED Dean Academics

Principal MIT Mysore

Probability Distributions : Theoretical distributions: Discrete and continuous random variables Discrete distributions: Geometric distributions, Hyper geometric distribution and Uniform distribution. Continuous distributions: Uniform Distribution, Gamma distributions, t-distribution, F-distribution and chi-square distribution	L1, L2, L3
Module -2	
Design of experiments: Analysis of variance, no way classification, completely Randomized design, randomized Block Design, Latin square design, Graeco Latin Squares	L1, L2, L3
Module -3	
ESTIMATION Parameter estimation-Point and interval; Estimation error-bias, variance and risk, Method of moments, Estimator design approach- Maximum Likelihood, confidence interval**.**	L1, L2, L3
Module -4	
Reliability Engineering: Concepts of Reliability, Reliability of systems, Availability of Markovian Systems Availability Function	L1, L2, L3
Module -5	
Stochastic Process: Classification of stochastic process with examples. Markov chain and related problems. Queuing theory- Poisson queuing system, Little law. Discussion of M/M/1 and M/M/s queuing models	L1, L2, L3

Text Books

1. K.F. Riley, M.P.Hobson and S.J. Bence, "Mathematical Methods for Physics and Engineering", Cambridge University Press 3rd Edition, 2017.
2. E. Kreyszig John Wiley & Sons, "Advanced Engineering Mathematics" 10thEd.,(Reprint), 2017.
3. T.Veerarajan, "Probability, Statistics and Random Process", Tata Mc-Graw Hill Co. 3rd Edition, 2016..

References Books

1. S. S. Sastry, "Introductory Methods of Numerical Analysis", Prentice Hall of India 4th Edition, 2011.
2. M. K. Jain, S. R. K. Iyengar and R. K. Jain, "Numerical Methods for Scientific and Engineering", Computation New Age Int. Publishers 6th Edition, 2014.
3. G.R. Grimmet and D.R. Stirzaker, "Probability and Random Processes", Oxford University Press 3rd Edition, 2001.
4. G. Haribaskaran "Probability, Queueing Theory and Reliability Engineering

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2: Probability Distributions:**	• Discrete and continuous random variables • Geometric distributions • Hyper geometric distribution and Uniform distribution • Continuous distributions: Uniform Distribution • Gamma distributions, t-distribution • F-distribution and chi-square distribution • Worked Problems
2	**Week 3-4: Design of experiments:**	• Analysis of variance, no way classification • completely Randomized design • Randomized Block Design • Latin square design • Graeco Latin Squares

		• Worked Problems
3	**Week 5-6: Estimation**	• Parameter estimation-Point and interval • Estimation error-bias, variance and risk • Method of moments • Estimator design approach • Maximum Likelihood, confidence interval • Worked Problems
4	**Week 7-8: Reliability Engineering**	• Concepts of Reliability • Reliability of systems • Worked Problems • Availability of Markovian • Systems Availability Function • Worked Problems
5	**Week 9-10: Stochastic Process**	• Classification of stochastic process with examples • Markov chain and related problems. • Queuing theory- Poisson queuing system • Little law and Problems • Discussion of M/M/1 queuing models. • Discussion of M/M/ queuing models.
6	**Week 11-12: Integration and Practical Applications**	• Apply learned concepts and competencies to real-world scenarios. Hands-on practice

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE Split up for Professional Elective Course (PE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	12.5
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	12.5
	Total Marks			**50**	**25**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examinations:

1. Question paper pattern will be 10 questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the 2 questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Probability Distributions**	Students will learn to calculate confidence intervals for parameters and to calculate critical regions for hypothesis tests. For univariate data, it is often useful to determine a reasonable distributional model for the data
2	**Design of experiments**	Students will learn to design of experiments, Block Design, Latin square design, Graeco Latin Squares
3	**Proficiency in Reliability engineering**	Students will become proficient in applied to many business functions, from design to maintenance.
4	**Project-Based Learning**	Through hands-on projects, students will apply their knowledge of Make use Estimations and Stochastic process
5	**Collaboration and Communication Skills**	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	**Ethical and Professional Responsibility**	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA206C.1	Apply the concepts of stochastic process Probability Distributions, estimation and design of experiments to solve the engineering problems
M23MCA206C.2	Demonstrate the importance of Probability Distributions, estimation and stochastic process in Computer Science Engineering
M23MCA206C.3	Analyze the Computer Science Engineering applications problems through probability, stochastic process

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA206C.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCA206C.2	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCA206C.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**

M23MCA206C	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	**Total**
Module 1	**2**	**5**	**3**	**10**
Module 2	**2**	**5**	**3**	**10**
Module 3	**2**	**5**	**3**	**10**
Module 4	**2**	**5**	**3**	**10**
Module 5	**2**	**5**	**3**	**10**
Total	**10**	**25**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	**Total**
Module 1	**4**	**10**	**6**	**20**
Module 2	**4**	**10**	**6**	**20**
Module 3	**4**	**10**	**6**	**20**
Module 4	**4**	**10**	**6**	**20**
Module 5	**4**	**10**	**6**	**20**
Total	**20**	**50**	**30**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

1. **Numerical analysis:** The course contributes to the understanding to solve continuous problems using numeric approximation. It involves designing methods that give approximate but accurate numeric solutions, which is useful in cases where the exact solution is impossible or prohibitively expensive to calculate
2. **Telecommunications and Networking**: Probability theory is essential in the design and analysis of communication systems, including wireless networks, telecommunications networks, and the internet. It helps in optimizing resource allocation, managing network congestion, and evaluating system performance.
3. **Mathematical Finance:** Probability as a subject in and of itself has rarely been truly appreciated by mathematicians in other disciplines. This has gradually changed over the last 50 years, as occasionally brilliant mathematicians' show how it can be used to solve, or to explain, and/or to give intuitive content to thorny mathematical issues. We provide some examples and then give a wild speculation as to where the field, at least in Mathematical Finance, might go in the future.

2nd Semester	PROFESSIONAL CORE CORSE LABORATORY (PCL) DATABASE LABORATORY	M23MCAL207

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Relational Data Base Management System**	• A solid understanding of how computers work, file management, and using software applications is essential. • Develop your analytical and problem-solving skills, as designing efficient and effective databases often requires making trade-offs and optimizing for different scenarios.

2. Competencies

S/L	Competency	KSA Description
1.	**Data Modeling**	**Knowledge:** Understand the principles of data modeling. **Skills:** Entity-Relationship diagrams (ERDs), **Attitudes:** These concepts help design efficient and organized database.
2.	**Relational Algebra and Set Theory**	**Knowledge:** Gain basic knowledge of relational algebra and set theory. **Skills:** The knowledge used to interact with relational databases. **Attitudes:** The foundation of relational databases.
3.	**SQL (Structured Query Language):**	**Knowledge:** the basics of SQL, the standard language for data query. **Skills:** Writing queries to retrieve, update, and manipulate data. **Attitudes:** Acquired skill to be used for querying with relational databases.
4.	**Normalization**	**Knowledge:** Learn about database normalization. **Skills:** To eliminate redundancy and improve data integrity. **Attitudes:** Understand the concept of normalization for optimizing query performance.
5.	**DataBase applications**	**Knowledge:** Gain insight into query optimization strategies. **Skills:** To design data base structure for aparticular application. **Attitudes:** To enhance database performance.

3. Syllabus

<table>
<tr><th colspan="4">DATABASE LABORATORY
SEMESTER – II</th></tr>
<tr><td>Course Code</td><td><b>M23MCAL208</b></td><td>CIE Marks</td><td><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week(L: T: P: S)</td><td><b>(0:0:3:0)</b></td><td>SEE Marks</td><td><b>50</b></td></tr>
<tr><td>Credits</td><td><b>02</b></td><td>Exam Hours</td><td><b>03</b></td></tr>
<tr><td colspan="4"><b>Course objectives:</b>
• Create SQL queries for the small projects.
• Create database objects that include tables, constraints, indexes, and sequences.</td></tr>
</table>

Sl. No	Experiments
1	**Consider the schema for College Database:** ***STUDENT(USN, SName, Address, Phone, Gender)*** ***SEMSEC(SSID, Sem, Sec)*** ***CLASS(USN, SSID)*** ***COURSE(Subcode, Title, Sem, Credits)*** ***IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)*** **Write SQL queries to :** 1. List all the student details studying in fourth semester 'C' section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN '4MH22CS200' in all Courses. 4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

Dean Academics
MIT Mysore

APPROVED
Dean Academics

Principal
MIT Mysore

	5. Categorize students based on the following criterion: *If FinalIA = 17 to 20 then CAT = ‘Outstanding’* *If FinalIA = 12 to 16 then CAT = ‘Average’* *If FinalIA< 12 then CAT = ‘Weak’* Give these details only for 4th semester ALL section students.
2	**Consider the schema for Movie Database:** ***ACTOR(Act_id, Act_Name, Act_Gender)*** ***DIRECTOR(Dir_id, Dir_Name, Dir_Phone)*** ***MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)*** ***MOVIE_CAST(Act_id, Mov_id, Role)*** ***RATING(Mov_id, Rev_Stars)*** **Write SQL queries to** 1. List the titles of all movies directed by ‘Mani Rathnam’. 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and also in a movie after 2024(use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed by ‘Shankar’ to 5.
3	**Consider the schema for Bus ticket reservation Database:** ***PASSENGER (P_id, P_Name, P_Gender,P_city)*** ***AGENCY(A_id,A_Name,A_city)*** ***BUS(B_id,B_date, B_time, B_scr, B_dest)*** ***BOOKING(P_id, A_id,B_id,B_date,B_time)*** Write SQL queries to 1. Get the Complete Details of all the Buses to MIT-Mysore. 2. Find only the Bus Number for Passenger with PID 123 for buses to Mandya, before 05/05/2024. 3. Find the Passenger Name for those who don’t have any booking in any buses. 4. Get the Details of the buses that are scheduled on both dates 01/02/2024 and 02/02/2024 at 16:00 hrs. Find the details of male passengers who are associated with “Happy Smiling” Agency.
4	***Consider the schema for Employee salaryDatabase:*** ***EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo)*** ***DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate)*** ***DLOCATION(DNo,DLoc)*** ***PROJECT(PNo, PName, PLocation, DNo)*** ***WORKS_ON(SSN, PNo, Hours)*** Write SQL queries to 1. Retrieve the employee numbers of all employees who work on project located in Mysore, Hassan, or Mangalore 2. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive) 3. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with all the details of Employee. 4. Select the names of employees whose salary is greater than the average salary of all employees in department 10. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.

5	**Consider the schema for MatrimonialDatabase :**
	ENROLL(E_Name,E_Gender,E_Age,E_Qualification, E_Salary,E_Address, E_ City)
	WORKS(E_name, E_salary, E_city)
	MIRRAGE_BUREAU(MB_name,MB_city,MB_charge)

	Write SQL queries to : 1. Find the Names and Cities for allthe Groom, who work for MIT-Mysore and earn more than Rs.60,000/- as salary. 2. Find the Company that has the Least Fee for Marriage Service. 3. Find the name of all the Brides in the database who live in the same cities and on the same street as do their Groom. 4. Find the names of Groom in the database, whose qualification and age is same as bride. Fine the name of the Groom in the database, who earns more than all Bride lives in "Mandya".
Demonstration Experiments (For CIE only – not to be included for SEE)	
7	Hospital Database Management system.
8	Timetable allotment and scheduling system.
9	E-commerce database management system

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-3 ER diagram and concepts**	Understand the principles of data modeling Entity-Relationship diagrams (ERDs). These concepts help design efficient and organized database.
2	**Week 4-6 RDBMs program implementation**	Gain basic knowledge of relational algebra and set theory. The knowledge used to interact with relational databases and the foundation of relational databases.
3	**Week 7-9 SQL programming**	The basics of SQL, the standard language for data query Writing queries to retrieve, update, and manipulate data.
4	**Week 10-12 Implement the concepts of Normalization**	Learn about database normalization to eliminate redundancy and improve data integrity. Understand the concept of normalization for optimizing query performance.
5	**Week 13-15 Database application**	Gain sight into query optimization strategies to enhance database performance. To design data base structure for a particular application.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Using traditional lecture methods and ICT as and when needed.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance learning.
3	**Collaborative Learning**	Encourage collaborative learning approaches for peer learning.
4	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application.
5	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.

6. Assessment details

CIE for Practical Courses (Laboratory Based):

- CIE marks for a practical course shall be 50 marks.
- The split up of CIE marks for record/journal and test to be split in the ratio 60:40
- Record write up for individual program/experiment will be evaluated for 10 Marks
- Total marks scored for record writing and conduction shall be scaled downed to 30 marks **(60% of the CIE Lab Marks (50))**

- 1 (one) test for 100 marks after the completion of the experiments at the end of the semester. The Test marks should be scaled down to 20marks **(40% of the CIE Lab Marks (50))Test**

Marks distribution for Laboratory based Practical Course for TEST

Sl. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	60
2	Viva-Voce	40%	40
Total		**100%**	**100**

Final CIE in Practical Course:

Marks distribution for Laboratory based Practical Course for Final CIE

Sl. No.	Description	% of Marks	In Marks
1	Scaled Down marks of Record	60% of the maximum	30
2	Scaled Down marks of Test	40% of the maximum	20
Total		**100%**	**50**

SEE for Practical Course (Laboratory based):

Marks distribution for Laboratory based Practical Course for Final SEE

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

1. SEE marks for practical course shall be 50 marks
2. See for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives
1	To provide a strong foundation in database concepts, technology, and practice.
2	To practice SQL programming through a variety of database problems.
3	To understand the relational database design principles.
4	To design and build database application for real world problems.
5	To become familiar with database storage structures and access techniques.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs):

COs	Description
M23MCAL208.1	Understand and apply the basic elements of a relational database management system.
M23MCAL208.2	Apply various constraints, techniques and Structured Query Language (SQL) statement for database operations.
M23MCAL208.3	Analyze various database models and normalization for the given application.
M23MCAL208.4	Design and develop entity relationship model and database application.

CO-PO-PSO Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCAL208.1	3	-	-	-	-	-	-	-
M23MCAL208.2	3	-	-	-	-	-	-	-
M23MCAL208.3	-	3	-	-	-	-	-	-
M23MCAL208.4	-	-	3	-	-	-	-	-
M23MCAL208	3	3	3	-	-	-	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Laboratory Programs	**10**				**10**
		10			**10**
			15		**15**
				15	**15**
Total	**10**	**10**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Laboratory Programs	**20**				**20**
		20			**20**
			30		**30**
				30	**30**
Total	**20**	**20**	**30**	**30**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject:

- Data Organization and Storage: Companies can store their data in databases in a structured, organized manner, making it simpler to access and analyze.
- Data Analysis: Databases contain a lot of data, and with the correct tools, organizations can analyze that data to find insights that will help them make business decisions and strategies.
- Efficiency: Databases give companies a centralized area to keep their data, making it more straightforward for staff to retrieve the data they want, minimizing duplication of work and boosting efficiency.
- Security & Privacy: Databases let companies control who has access to their data, ensuring that only authorized users may see and change it. This aids in preventing unauthorized access to and breaches of vital consumer and corporate information.

2nd Semester	PROFESSIONAL CORE COURSE LABORATORY (PCL) JAVA PROGRAMMING LABORATORY WITH MINI PROJECT	M23MCAL208

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Computer Skills**	Basic computer skills, such as saving files in multiple versions and formats.
2	**Programming Fundamentals**	Familiar with general coding concepts like variables, basic data types, Conditional Statements, Looping, Functions, creation of source file, compilation process, program execution techniques.
3	**Multi-Process Execution Programming**	Familiar with the way to execute multiple tasks simultaneously by creating multiple processes.
4	**Basic Object Orientation Concepts**	Basic of four basic principles: encapsulation, inheritance, polymorphism, and abstraction. Where these four OOP principles can be used enable to create objects and collaborate to create powerful applications too.
5	**Programming basic tools**	Familiar with Programming tools like assemblers, compilers, linkers translate, flowchart, algorithms which can be used to form a program from a human write-able and readable source language into the bits and bytes that can be executed by a computer.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Object Oriented Concepts**	**Knowledge:** • Importance of Object Orientation Concepts. • Understanding of the basics of Object Orientation Programming. **Skills:** • Ability to apply Object Orientation Concepts to create objects using appropriate structure. **Attitudes:** • Appreciation to understand the importance of object orientation perspective and implement the same at basic level.
2	**Basic of Programming**	**Knowledge:** • Understanding of basic elements of programming specific to Java Language. • Basics of Java program execution. **Skills:** • Designing basic Java program using basic elements of programming language. • Creating and executing simple Java programs. **Attitudes:** • Appreciation for the role of Java programming elements and its execution.
3	**Java Classes and its methods**	**Knowledge:** • Understanding how classes are defined with data members and methods. **Skills:** • Designing of classes for real world objects. • Defining appropriate attributes and methods for classes. **Attitudes:** • Valuing the importance of classes and its methods in line with real-world objects.

4	**Reusability of Classes and Methods**	**Knowledge:** • Understanding the importance of code reusability through classes and methods reusability. **Skills:** • Applying concepts of object orientation with classes and methods. • Describing the actually importance of reusability through implementations. **Attitudes:** • Openness to learning and using object orientation concepts to achieve code reusability.
5	**Exceptions and Handling the Exceptions**	**Knowledge:** • Understanding of issues with exceptions. **Skills:** • Implementing how to handle the exceptions through appropriate Java programming construct. **Attitudes:** • Appreciation for the way exception is handled and making the execution of program in control.
6	**Multi-Threaded Programming**	**Knowledge:** • Understanding the characteristics and importance of parallel execution of a task. **Skills:** • Designing and analyzing the parallel execution using thread concepts. • Implementing multi-thread concepts. **Attitudes:** • Recognizing the significance of flip-flops in sequential logic circuits

3. Syllabus

JAVA PROGRAMMING LABORATORY WITH MINI PROJECT SEMESTER – II			
Course Code	**M23MCAL208**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:3:3:0)**	SEE Marks	**50**
Credits	**04**	Exam Hours	**03**

Course objectives:

- To learn primitive constructs JAVA programming language.
- To understand Object Oriented Programming Features of JAVA.
- To gain knowledge on: packages, multithreaded programing and exceptions.
- Experience the implementation by doing mini project on own.

Part A

1. Write a program to calculate salary of n employees using concept of classes with constructors and methods.
2. Write a program to demonstrate e-commerce website using inheritance, abstract class and dynamic polymorphism.
3. Write a program to demonstrate various arithmetic calculations using packages.
4. Write a program to demonstrate client-server environment using multithreading.
5. Write a program to demonstrate mutual exclusion using thread synchronization.
6. Write a program to demonstrate Hash set and Iterator classes.
7. Write a program to demonstrate Enumeration and Comparator interfaces.
8. Write a program to accept data and display output in key, value pair.
9. Write a program to create a registration form with different controls, menus and demonstrate event handling.
10. Write a program to copy data from one file to another file.
11. Write a program to merge contents of two files and display output on console.

12. Write a program to retrieve web page using URL class.

MINI PROJECT:

Implement mini project using all the Java concepts studied in the course of M23MCA202.

Following are some of the examples for Mini-projects:

(a) Railway reservation system
(b) Payroll management system
(c) Supermarket billing system
(d) Telephone directory system

Only one Mini-project is planned to be undertaken by a student that needs to be assigned to him/her.

Mini - Project Topic Selection, Approval, Report Writing and Evaluation :

1. The number of students per mini-project may be minimum THREE (03) and maximum FOUR (04).
2. Topic selection and approval by faculty from the Department.
3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It is recommended that students to do prior search as part of literature survey before submitting the synopsis for the Mini- projects
4. The team must submit a brief project report (20-25 pages) with following contents shall be prepared:
   - Title
   - Introduction
   - Scope of the work
   - Problem Statement
   - Selection of materials, calculations
   - Casting/Testing/Modelling Procedures
   - Results & Discussions
   - Conclusions
   - References
5. Mini – project assessment must be based on the overall performance of the student with every experiment graded/ Marks award from time to time.
6. The ‘Practical and Oral’ examination will be based on (a) the final project reports (maximum 05 marks), (b) projects presentation (maximum 05 marks), (c) demonstration of the projects(maximum 10 marks), and (d) questions and answers during Oral (maximum 05 marks)

NOTE:

Part A: The student should have experience implementing basic programming constructs like control structures, constructors, string handling, garbage collection and implementation of inheritance, Etc.

Part B: Each student has to execute one program picked from Part-A during the semester end examination. In CIE/SEE Part-A and Part-B shall be given 50% weightage each.

4. Syllabus Timeline

S/L	**Syllabus Timeline**	**Description**
1	Week 1-2: **Constructors implementation**	The student should have experience implementing basic programming constructs like control structures, constructors, string handling, garbage collection
2	Week 3-4: **Inheritance, polymorphism implementation**	Demonstrate e-commerce website using inheritance, abstract class and dynamic polymorphism
3	Week 5-6: **Arithmetic calculations and Threading implementation**	Implementation of various arithmetic calculations using packages. Demonstrate client-server environment using multithreading.

4	Week 7-8: **Event handling** **Files, Web page implementation**	Creating different controls, menus and demonstrate event handling. Demonstrating of copying data from one file to another file. How to retrieve web page using URL class
5	Week 9-10: **Implement mini project using all the Java concepts**	Mini-project is planned to be undertaken by a student that all the JAVA concepts are implemented and student will experience all the features of the programming concepts.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Image/Video/Animation**	Incorporate visual aids like image/videos/animations to enhance understanding of programming constructs.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Programming-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

CIE for Practical Courses (Laboratory Based):

- ➢ CIE marks for a practical course shall be 50 marks.
- ➢ The split up of CIE marks for record/journal and test to be split in the ratio 60:40
- ➢ Record write up for individual program/experiment will be evaluated for 10 Marks
- ➢ Total marks scored for record writing and conduction shall be scaled downed to 30 marks **(60% of the CIE Lab Marks (50))**
- ➢ 1 (one) test for 100 marks after the completion of the experiments at the end of the semester. The Test marks should be scaled down to 20marks **(40% of the CIE Lab Marks (50))Test**

Marks distribution for Laboratory based Practical Course for TEST

Sl. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	60
2	Viva-Voce	40%	40
	Total	**100%**	**100**

Final CIE in Practical Course:

Marks distribution for Laboratory based Practical Course for Final CIE

Sl. No.	Description	% of Marks	In Marks
1	Scaled Down marks of Record	60% of the maximum	30
2	Scaled Down marks of Test	40% of the maximum	20
	Total	**100%**	**50**

SEE for Practical Course (Laboratory based):

- SEE marks for practical course shall be 50 marks

Marks distribution for Laboratory based Practical Course for Final SEE

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

- See for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
- Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
- Duration of SEE shall be 3 hours.

Mini Project:

Mini Project shall be evaluated as per the following guidelines

- The CIE marks awarded for mini project shall be based on the evaluation of mini-project work by the guide, report writing and viva-voce in the ratio 50:25:25.
- Marks awarded for the mini project report shall be based on the performance of the students of the batch.

  The guide shall evaluate the performance for 50% of the maximum marks of CIE for the report, 25% for presentation and 25% for viva-voce.

Mini Project Evaluation for CIE

SL.No.	Description	% of Marks	In Marks
1	Mini Project Report	50%	50
2	Presentation Skills	25%	25
3	Viva-Voce	25%	25
Total		**100%**	**100**

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding basic Java Programming**	Students will grasp the fundamental concepts of Java Programming, including basic constructs.
2	**Designing simple Programs**	Students will learn to design and implement basic and simple Java programs.
3	**Proficiency in Java**	Students will become proficient in understanding and applying the Java specific constructs to improve the efficiency of Java programming logics.
4	**Programming-Based Learning**	Through program execution-based learning, students will undergo the demonstration of Java programming constructs working principles.
5	**Ethical and Professional Responsibility**	Students will understand the ethical and professional responsibilities associated with Java Programming, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCAL208.1	Understand and apply the basic programming constructs.
M23MCAL208.2	Apply the structure of classes and methods in Java programming environment.
M23MCAL208.3	Analyze the different programming constructs of Java and its effectiveness in improving the efficiency of Java programs.
M23MCAL208.4	Implement appropriate Java programming constructs to solve real-world problem sample scenarios.

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCAL208.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCAL208.2	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCAL208.3	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**
M23MCAL208.4	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**
M23MCAL208	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Laboratory Programs	**10**				**10**
		10			**10**
			15		**15**
				15	**15**
Total	**10**	**10**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Laboratory Programs	**20**				**20**
		20			**20**
			30		**30**
				30	**30**
Total	**20**	**20**	**30**	**30**	**100**

10. Future with this Subject

The future of Java programming applications looks promising due to several factors:

1. **Enterprise Usage**: Java continues to be a go-to language for large-scale enterprise applications, thanks to its stability, scalability, and robust security features.
2. **Android Development**: Java remains a key language for Android app development, although Kotlin is becoming increasingly popular.
3. **Evolving Ecosystem**: The Java ecosystem, including frameworks like Spring and tools like Maven and Gradle, continues to evolve, making development more efficient and powerful.
4. **Cloud and Big Data**: Java's performance and reliability make it a good fit for cloud computing and big data applications, areas that are rapidly growing.
5. **Community and Support**: Java has a large, active community and strong support from industry giants like Oracle, ensuring ongoing development and support.

In simple terms, Java is likely to remain a critical technology for business applications, mobile development, and emerging tech fields like cloud computing and big data.

2nd Semester	**MANDATORY CREDIT COURSE (MC) PROFESSIONAL COMMUNICATION AND SKILL ENHANCEMENT -2**	**M23MCA209**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Reading and Writing Skills**	Ability to read and comprehend texts, and write clearly and coherently.
2	**Critical Thinking**	Willingness to analyze situations, identify patterns, and think critically about solution
3	**Basic Language Skills**	Good comprehension and basic grammar skills for understanding and interpreting verbal reasoning questions
4	**Attention to Detail**	Careful and precise attention to details, important for identifying nuances in questions and data
5	**Open Mindset**	Openness to learning new strategies and techniques for approaching different types of logical and aptitude problems

2. Competencies

S/L	Competency	KSA Description
1.	**Basic Math Skills**	**Knowledge:** • Familiarity with logical reasoning principles, such as patterns, sequences, and relationships. **Skills:** • Ability to sole the simple logical problems **Attitudes:** • Analyzing the given problem and apply suitable logic
2.	**Problem-Solving Skills**	**Knowledge:** • Email, Resume Writing, Online Communication **Skills:** • Ability to approach and solve problems systematically and logically **Attitudes:** • A strong desire to learn and understand new concepts and solve challenging problems.
3.	**Analytical Thinking**	**Knowledge:** • Ability to break down complex problems in to simpler components to understand and solve them **Skills:** • Pro Skill in systematically analyzing problems to determine the best solution **Attitudes:** • Belief in one's own abilities to solve problems and tackle new challenges.
4.	**Professional Communication**	**Knowledge:** • Importance, Basics, purpose & audience, cross cultural communication, Language as a tool. **Skills:** • Controlling nervousness & stage Fright, Visual aids in presentation. **Attitudes:** • Classification of barriers, Effective Presentation Strategies.
5.	**Verbal Reasoning**	**Knowledge:** • Understanding language-based logic and the ability to comprehend and analyze written information.

		Skills: • Ability to clearly explain reasoning and solutions to others, both verbally and in writing. **Attitudes:** • Perform in a team to make an effective oral/written presentation.

3. Syllabus

PROFESSIONAL COMMUNICATION AND SKILL ENHANCEMENT -2
SEMESTER–II

Course Code	**M23MCA209**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T:P:S)	**(2:0:0:2)**	SEE Marks	**50**
Total Hours of Pedagogy	**36 hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**01**

Course objectives:

- Learn and inculcate concepts of Professional Communication and Ethics.
- Acquire knowledge about logical reasoning and problem solving.

Module-1

Logical Aptitude -Syllogism, Venn-diagram method, Three statement syllogism, Deductive and inductive reasoning. Introduction to puzzle and games organizing information, parts of an argument, common flaws, arguments and assumptions.
Linear Seating Arrangement: Single or Double rows facing each other or away from each other in the same direction Circular Seating Arrangement**.** Uni-&Bi-directional problems on Circular, Square, Rectangular, Hexagonal tables
Coding Decoding: Letter Coding, Number Coding, symbol coding **Crypt arithmetic:** Basic concepts, addition, subtraction, multiplication of coded alphabets, Types of cryptarithm Clocks and Calendar.
Reasoning–Verba l- Blood Relation, Sense of Direction, Arithmetic & Alphabet. Non-Verbal reasoning-Visual Sequence, Visual analogy and classification.
Analytical Reasoning-Single & Multiple comparisons, Linear Sequencing.

L1, L2

Module-2

Ratio and Proportion: Simple Ratios**,** Compound Ratios**,** Comprehend and Dividend**,** Direct & Indirect Proportions**,** Problems on ages
Mixtures & Allegation: Speed, Time and Distance**,** Relative Speed**,** Average Speed**,** Problems on Train**,** Boat & Stream. Time and Work**,** Work Efficiency**,** Work & Wages Pipes & Cisterns
Permutation and Combination: Understanding the difference between the permutation and combination, Rules of Counting-rule of addition, rule of multiplication, factorial function, Concept of step arrangement, Permutation of things when some of them are identical, Concept of 2n, Arrangement in a circle.
Probability: Single event probability, multi event probability, independent events and dependent events, mutually exclusive events, non-mutually exclusive events, combination method for finding the outcomes.
Coding Decoding: Letter Coding, Number Coding, symbol coding Crypt arithmetic: Basic concepts, addition, subtraction, multiplication of coded alphabets, Types of cryptarithm
Progression: Arithmetic Progression, sum of given number of terms in an A.P., arithmetic mean, to insert a given number of arithmetic means between two given quantities, nth term of an A.P., finding common difference of an A.P.given2termsof an A.P., types of A.P.s–increasing A.P.s and decreasing A.P.s
Geometric: to find, the geometric mean between two given quantities, to insert a given number of geometric means between two given quantities, sum of a number of terms in a G.P. Types of G.P.s—increasing G. P. s type one and two, decreasing G. P. s type one and two. Harmonic Progression: to find the harmonic mean between two given quantities , theorems related with progressions, solved examples sample company questions

L1, L2

Data Interpretation: Approach to interpretation-simple arithmetic, rules for comparing fractions, calculating (approximation) fractions, short cut ways to find the percentages, Classification of data–Tables, Bar graph, line graph, Cumulative bar graph, Pie graph, Combination of graphs. Combination of table and graphs.	
Module-3	
Identifying Common Errors in writing and Speaking English: Advanced English Grammar for Professionals with exercises, Common errors identification in parts of speech, Use of verbs and phrasal verbs, Auxiliary verbs and their forms, Subject Verb Agreement (Concord Rules with Exercises).Common errors in Subject-verb agreement, Noun-pronoun agreement, Sequence of Tenses and errors identificationinTenses.AdvancedEnglishVocabularyanditstypeswithexercises–VerbalAnalogies, Words Confused/Misused.	**L2, L3**
Module-4	
Technical Reading and Writing Practices: Reading Process and Reading Strategies, Introduction to Technical writing process, understanding of writing process, Effective Technical Reading and Writing Practices, Introduction to Technical Reports writing, Significance of Reports, **Types of Reports:** Introduction to Technical Proposals Writing, Types of Technical Proposals, Characteristics of Technical Proposals. Scientific Writing Process. **Grammar**–Voice and Speech (Active and Passive Voices) and Reported Speech, Spotting Error Exercises, Sentence Improvement Exercises, Cloze Test and Theme Detection Exercises. **Nature and Style of sensible writing:** Organizing Principles of Paragraphs in Documents, Writing Introduction and Conclusion, Importance of Proper Punctuation, The Art of Condensation (Precise writing) and Techniques in Essay writing, Common Errors due to Indianism in English Communication, Creating Coherence and Cohesion, Sentence arrangements exercises, Practice of Sentence Corrections activities. Importance of Summarizing and Paraphrasing. Misplaced modifiers, Contractions, Collocations, Word Order, Errors due to the Confusion of words, Common errors in the use of Idioms and phrases, Gender, Singular & Plural, Redundancies & Clichés.	**L2, L3**
Module-5	
Business Etiquettes: Greetings and Introductions in Business Settings**,** Business Dining Etiquette, Dress Code and Personal Grooming**,** Electronic Etiquette: Phone, Email, and Social Media **International Business Etiquette:** Understanding Cultural Differences Work Ethic and Professionalism, Defining Work Ethic: Traits and Characteristics, The Importance of Reliability and Accountability, Maintaining Confidentiality, Building a Positive Professional Image, Balancing Professionalism with Personal Authenticity	**L3, L4**

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week1-2:** **Logical Aptitude** **Analytical Reasoning**	• Logical Aptitude • Solving problems logically • Analytical Reasoning
2	**Week3-4:** **Ratio and Proportion** **Permutation and Combination**	• Permutation and Combination • Solving various probability problems • Data Interpretation: Approach to interpretation
3	**week5-6** **Identifying Common Errors in writing and Speaking English**	• Advanced English Grammar for Professionals • Common errors identification in parts of speech, Use of verbs and phrasal verbs, Auxiliary verbs • Advanced English Vocabulary and its types with exercises

4	**Week7-8: Technical Reading and Writing Practices**	• Technical Reading and Writing Practices • Nature and Style of sensible writing • Grammar, Parts of Speech, Importance of Summarizing and Paraphrasing.
5	**Week9-10: Business Etiquettes International Business Etiquette**	• Business Etiquettes • Understanding Cultural Differences Work Ethic and Professionalism • Balancing Professionalism with Personal Authenticity

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**Higher Order Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Real-World Application**	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	**Flipped Class Technique**	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	**Programming Assignments**	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details

Assessment Details (both CIE and SEE)

Scheme of Continuous Internal Examination (CIE): Evaluation of CIE will be carried out in TWO Phases.	
Phase	**Activity**
I	CIE1 is conducted for 30 marks is consolidated to 20 Marks.
II	CIE2 is conducted for 30 marks is consolidated to 20 Marks.
III	CIE1 (20 marks) + CIE2 (20marks) + Attendance (10 marks) = 50 marks10 marks for attendance will be considered only if students have more than 85% attendance
IV	SIE is conducted for 50 marks (Students are allowed to write SIE provide they have minimum of 50% CIE marks and more than 85% attendance

7. Learning Objectives

S/L	Objectives	Description
1	**Critical Thinking**	Develop the ability to analyze problems logically and make well-reasoned decisions.

2	**Problem-Solving Skills**	Learn to approach and solve various types of problems systematicallyand efficiently.
3	**Logical Reasoning**	Understand and apply principles of logic to deduce conclusions fromgiven premises.
4	**Quantitative Aptitude**	Gain proficiency in basic mathematical concepts and numericalcalculations.

8. Course Outcomes (COs) and Mapping with POs

Course Outcomes (COs)

COs	Description
M23MCA209.1	Logical reasoning and aptitude are to develop critical thinking, problem-solving skills, and the ability to analyze and interpret data effectively. Also enhance professional communication skills

CO-PO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
M23MCA209.1	**2**	**2**	**2**	**-**	**2**	**-**	**-**	**3**
M23MCA209	**2**	**2**	**2**	**-**	**2**	**-**	**-**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	Total
Module 1	**6**	**6**
Module 2	**6**	**6**
Module 3	**6**	**6**
Module 4	**6**	**6**
Module 5	**6**	**6**
Total	**30**	**30**

Semester End Examination (SEE)

	CO1	Total
Module 1	**10**	**10**
Module 2	**10**	**10**
Module 3	**10**	**10**
Module 4	**10**	**10**
Module 5	**10**	**10**
Total	**50**	**50**

10. Future with this Subject

The future of logical reasoning and aptitude courses is adapting to new challenges and opportunities.

- **Critical Thinking**: Emphasis on enhancing critical thinking skills to tackle complex problems effectively.
- **Digital Tools**: Using technology and digital platforms to improve learning experiences and practice.
- **Real-World Relevance**: Applying logical reasoning and aptitude skills to real-world scenarios andpractical situations.
- **Data Analysis**: Teaching data interpretation and analysis, which are increasingly valuable skills.

- **Interdisciplinary Approach**: Combining reasoning skills with knowledge from various fields forcomprehensive problem-solving
- **Continuous Learning**: Encouraging ongoing development and updating of skills to stay relevant in afast-changing world.
- **Soft Skills Integration**: Blending logical reasoning with soft skills like communication, creativity,and teamwork.

In simple terms, the future of logical reasoning and aptitude courses is about developing critical thinking, leveraging technology, applying skills to real-life situations, and integrating soft skills for well-rounded problem-solving abilities.