SAMARTH EDUCATIONAL TRUST

ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA

(AN AUTONOMOUS INSTITUTE)

Curriculum for First Year M.C.A.

List of Abbreviations

Sr. No.	Abbreviation	Description	Code
1	L	Lecture	
2	T	Tutorial	
3	P	Practical	
4	Cr	Credits	
5	BSC	Basic Science Course	BS
6	ESC	Engineering Science Course	ES
7	AEC	Ability Enhancement Course	AE
8	VSEC	Vocational and Skill Enhancement Course	VS
9	PCC	Program Core Course	PC
10	IKS	Indian Knowledge System	IK
11	CC	Co-curricular Course	CC

SAMARTH EDUCATIONAL TRUST
ARVIND GAVALI COLLEGE OF ENGINEERING
• ENGINEERING (B.Tech & M.Tech) • BCA • MCA • B.VOC
• NAAC & NBA Accredited • AN AUTONOMOUS INSTITUTE • ISO 9001:2015
Approved by AICTE, New Delhi, Recognised by Govt. Of Maha.,DTE Mumbai & Affiliated to
Dr.Babasaheb Ambedkar Technological University (BATU), Lonere.

• Address : At.Panmalewadi, Post.-Varye, Tal.& Dist.-Satara.-415 015 (Maharashtra)
• Phone : 02162 - 261122 , 200100
• e-mail : agcenggsatara@gmail.com
• Website :-www.agce.edu.in
• Institute Code : Engg. DTE EN-6545
• BCA 6545, MCA 6545, B.Voc 6545

Ref No.: Date :

As per NEP Guidelines, Proposed Scheme of Credit Distribution
First Year MCA 2025-26
SEMESTER I

Sr. No.	Category	Course Code	Course Name	Teaching Scheme					Evaluation Scheme			
				L	T	P	Hrs. /Wk	Cr	Components	Max	Min for Passing	
1	PCC	25MCA1101	Advance Operating System	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PCC	25MCA1102	Data Structure and Algorithms using python	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
3	PCC	25MCA1103	Data Base Management System	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
4	BSC	25MCA1104	Probability and Statistics	3	1	0	4	4	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
5	PCC	25MCA1105	Java Programming	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
6	PCC	25MCA1102L	Data Structure & Algorithms using Python Laboratory	0	0	2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
7	PCC	25MCA1103L	Data Base Management System Laboratory	0	0	2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
8	PCC	25MCA1105L	Java Programming Laboratory	0	0	2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
9	VSEC	25MCA1106L	Web Technology Laboratory	0	0	2	2	1	CA1	25		20
									CA2	25		
									POE	--	--	
			Total	**15**	**1**	**8**	**24**	**20**		**850**		
Total Contact Hours –24 Total Credits – 20												

SAMARTH EDUCATIONAL TRUST
ARVIND GAVALI COLLEGE OF ENGINEERING
• ENGINEERING (B.Tech & M.Tech) • BCA • MCA • B.VOC
• NAAC & NBA Accredited • AN AUTONOMOUS INSTITUTE • ISO 9001:2015
Approved by AICTE, New Delhi, Recognised by Govt. Of Maha.,DTE Mumbai & Affiliated to Dr.Babasaheb Ambedkar Technological University (BATU), Lonere.

• Address : At.Panmalewadi, Post.-Varye, Tal.& Dist.-Satara.-415 015 (Maharashtra)
• Phone : 02162 - 261122 , 200100
• e-mail : agcenggsatara@gmail.com
• Website :-www.agce.edu.in
• Institute Code : Engg. DTE EN-6545
• BCA 6545, MCA 6545, B.Voc 6545

Ref No.: Date :

SEMESTER II

Sr. No.	Category	Course Code	Course Name	Teaching Scheme					Evaluation Scheme			
				L	T	P	Hrs. / Wk	Cr	Components	Max	Min for Passing	
1	PCC	25MCA1201	Object Design & Agile Development	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PCC	25MCA1202	Computer Networks	3	1	0	4	4	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
3	PCC	25MCA1203	Artificial Intelligence	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
4	PEC	25MCA1204	Program Elective I	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
5	OE	25MCA1205	Open Elective I	3	0	0	3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
6	PCC	25MCA1201L	Object Design & Agile Development Laboratory	0	0	2	2	1	CA1	25		20
									CA2	25		
									POE	--	--	
7	VSEC	25MCA1206L	Data Analysis with Python Laboratory	0	0	2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
8	PCC	25MCA1203L	Artificial Intelligence Laboratory	0	0	2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
9	Project	25MCA1207L	Mini Project	0	0	2	2	1	CA1	25		40
									CA2	25		
									OE	50	20	
10	Internship	25MCA1208	Field Training / Internship / Industrial Training Evaluation					AU				
			Total	**15**	**1**	**8**	**24**	**20**		**850**		
Total Contact Hours – 24 Total Credits – 20												

SAMARTH EDUCATIONAL TRUST
ARVIND GAVALI COLLEGE OF ENGINEERING
• ENGINEERING (B.Tech & M.Tech) • BCA • MCA • B.VOC
• NAAC & NBA Accredited • AN AUTONOMOUS INSTITUTE • ISO 9001:2015
Approved by AICTE, New Delhi, Recognised by Govt. Of Maha.,DTE Mumbai & Affiliated to Dr.Babasaheb Ambedkar Technological University (BATU), Lonere.

• Address : At.Panmalewadi, Post.-Varye, Tal.& Dist.-Satara.-415 015 (Maharashtra)
• Phone : 02162 - 261122 , 200100
• e-mail : agcenggsatara@gmail.com
• Website :-www.agce.edu.in
• Institute Code : Engg. DTE EN-6545
• BCA 6545, MCA 6545, B.Voc 6545

Ref No.: Date :

Semester III

Sr. No.	Category	Course Code	Course Name	Teaching Scheme					Evaluation Scheme			
				L	T	P	Hrs./ Wk	Cr	Components	Max	Min for Passing	
1	PCC	25MCA2101	Full Stack Development	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PCC	25MCA2102	Machine Learning	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
3	PCC	25MCA2103	Cloud Computing	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
4	PEC	25MCA2104	Program Elective II	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
5	PEC	25MCA2105	Program Elective III	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
6	AEC	25MCA2106	Innovation, Business Models and Entrepreneurship	2			2	2	CA1	25		20
									MSE	--		
									CA2	25		
									ESE	--	--	
7	PCC	25MCA2101L	Full Stack Development Laboratory			4	4	2	CA1	25		45
									CA2	25		
									POE	50	20	
8	PCC	25MCA2102L	Machine Learning Laboratory			2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
10	Project	25MCA2107L	Project phase –I			2	2	1	CA1	25		45
									CA2	25		
									POE	50	20	
			Total	**17**	**0**	**8**	**25**	**21**		**850**		

Total Contact Hours –25 Total Credits – 21

SAMARTH EDUCATIONAL TRUST
ARVIND GAVALI COLLEGE OF ENGINEERING
• ENGINEERING (B.Tech & M.Tech) • BCA • MCA • B.VOC
• NAAC & NBA Accredited • AN AUTONOMOUS INSTITUTE • ISO 9001:2015
Approved by AICTE, New Delhi, Recognised by Govt. Of Maha.,DTE Mumbai & Affiliated to
Dr.Babasaheb Ambedkar Technological University (BATU), Lonere.

• Address : At.Panmalewadi, Post.-Varye, Tal.& Dist.-Satara.-415 015 (Maharashtra)
• Phone : 02162 - 261122 , 200100
• e-mail : agcenggsatara@gmail.com
• Website :-www.agce.edu.in
• Institute Code : Engg. DTE EN-6545
• BCA 6545, MCA 6545, B.Voc 6545

Ref No.: Date :

Semester IV

Sr. No.	Category	Course Code	Course Name	Teaching Scheme					Evaluation Scheme			
				L	T	P	Hrs./ Wk	Cr	Components	Max	Min for Passing	
1	PCC	25MCA2201	Software Engineering & Project Management	4			4	4	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PEC	25MCA2202	Program Elective IV	3			3	3	CA1	10		45
									MSE	30		
									CA2	10		
									ESE	50	20	
3	Internship/ OJT/ Project	25MCA2203	Project phase – II (In-house) / Internship and Project in the Industry			24	24	12	CA1	25		45
									CA2	25		
									OE	50	20	
			Total	**7**		**24**	**31**	**19**		**300**		
Total Contact Hours –26 Total Credits – 19												

Semester I

Title of the Course: Advance Operating System Course Code: 25MCA1101	L	T	P	Credit
	3	-	-	3

Course Prerequisite: Students should have a basic understanding of operating systems, computer architecture, and programming fundamentals in C or Java. Basic knowledge of data structures and algorithms (e.g., stacks, queues, linked lists, trees).

Course Description:
This course provides a comprehensive introduction to the fundamental concepts, design principles, and structures of modern operating systems. It covers key topics including process management, CPU scheduling, synchronization, memory management, file systems, and storage systems. The course also explores advanced topics such as virtual memory, deadlocks, distributed file systems (e.g., HDFS), and cloud storage solutions. Through theoretical study and practical insights, students will gain an understanding of how operating systems manage hardware and software resources efficiently and securely.

Course Objectives:
By the end of this course, students will be able to:
1. Understand the structure, functions, and services of operating systems.
2. Analyze process scheduling, synchronization, and deadlock handling techniques.
3. Apply concepts of memory management, paging, segmentation, and virtual memory.
4. Examine file systems, disk scheduling, and I/O management methods.
5. Implement system-level programs using operating system principles.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Grasp concepts, types, and structures of operating systems.
CO2	Utilize process scheduling and facilitate inter-process communication.
CO3	Execute synchronization techniques (e.g., semaphores).
CO4	Evaluate memory management strategies (paging, virtual memory).
CO5	Enhance techniques for managing files and disks.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2	3	2	2		3							
CO3	3	2	3		3							
CO4	3	3	3	2								
CO5	2		3	2	3							

Assessment Scheme:
Two components of Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and **CA2** are based on Surprise test/ Assignment/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content.
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	**Introduction and Operating system structures** Definition, Types of Operating system, Real-Time operating system, System Components: System Services, Systems Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generations.	**8**
2	**Processes and CPU Scheduling** Process Concept, Process Scheduling, Operation on process, Inter-process Communication, Cooperating processes, Threads, Multithreading model, Scheduling criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Scheduling Algorithms evaluation.	**8**
3	**Process Synchronization** The critical-section problem, Critical regions, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of synchronization, and Monitors Deadlocks: Systems Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Combined approach to deadlock Handling.	**8**
4	**Memory Management** Basic concept, Logical and Physical address map, Memory allocation: Continuous Memory Allocation, Fixed and variable partition, Internal and external fragmentation and compaction, Paging: Principle of operation, Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging; Segmentation. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page / Dirty bit – Demand paging, Page Replacement algorithms: Optimal, first in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used(LRU). Sequences and convergence, Infinite	**8**

	series, tests of convergence (comparison test, ratio test), Power series: Radius and interval of convergence, Taylor and Maclaurin series (with examples), Errors and approximations using Taylor's expansion (first-order).	
5	**File Management** File Concept, Access methods, File types, File operation, Directory and disk structure, File System Structure, File System Implementation, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Mass-Storage Structure: Disk Structure, Disk attachment, Disk scheduling, Disk management, Swap Space Management. Distributed File Systems:Introduction to Hadoop Distributed File System (HDFS),Architecture, Block Storage, DataNode & NameNode Concepts,Cloud Storage Solutions: Overview of Cloud Storage.	**8**

Text Books

Sr. No.	Title	Author	Publisher
1	Operating System Concepts	Abraham Silberschatz, Peter B.Galvin and Greg Gagne	8th Edition, 2008

Reference Books

Sr. No.	Title	Author	Publisher
1	Modern Operating System	Andrew S. Tanenbaum	4th Edition, 2015
2	Systems Programming and Operating Systems	D. M. Dhamdhere	2nd Edition, 1996
3	Operating Systems Concepts	Garry Nutt	3rd Edition, 2003
4	An Introduction to Operating Systems	Harvey M. Deitel	2nd Edition,1990
5	Operating System in Depth: Design and Programming	Thomas W. Doeppner	Wiley Publication, 2011

Title of the Course: Data Structure and Algorithms using Python Course Code: 25MCA1102	L	T	P	Credit
	3	-	-	3

Course Prerequisite: Basic understanding of programming concepts (variables, operators, control flow), Familiarity with problem-solving techniques, Exposure to any programming language (preferably Python or C) at an introductory level.

Course Description:

This course introduces the fundamental concepts of data structures and their implementation using Python. Students will learn about linear and non-linear data structures, their operations, and their applications in problem-solving. The course emphasizes both theoretical understanding and practical implementation, enabling students to write efficient Python programs for common data structures such as arrays, stacks, queues, linked lists, and trees, as well as apply algorithmic techniques like divide and conquer, greedy methods, and dynamic programming.

Course Objectives:

By the end of this course, the students will be able to:

1. Introduce data structures, abstract data types, and their use in problem-solving.
2. Implement and work with data structures in Python.
3. Analyze algorithms and select appropriate structures
4. Understand common algorithm design paradigms.
5. Enhance programming and logical thinking skills.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Grasp fundamental concepts of data structures and algorithmic problem solving using Python.
CO2	Implement linear and non-linear data structures such as stacks, queues, linked lists, trees, and graphs.
CO3	Apply searching and sorting algorithms effectively for problem solving.
CO4	Analyze the time and space complexity of algorithms using Python implementations.
CO5	Develop Python-based solutions for real-world computational problems using suitable data structures and algorithms.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3				2							
CO2	3	3	3		3							
CO3	3		2		2							
CO4	3	3		3								
CO5	3	2	2	2	3							

Assessment Scheme:
Two components of Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and **CA2** are based on Surprise test/ Assignment/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content.
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	**Python for Data Structures** Introduction to Python – features, comparison with C, Python syntax basics (variables, data types, operators, input/output). Control Structures: conditional statements (if, elif, else), loops (for, while) with range. Python Data Structures: lists, tuples, sets, dictionaries – creation, access, modification, traversal, basic operations (insertion, deletion, searching). Functions in Python – defining and calling functions, parameters, return values, simple examples for addition/searching. Implementing stack using list, queue using collections.deque, with examples.	**8**
2	**Introduction to Data Structure** Data, Data types, Data structure, Abstract Data Type (ADT), representation of Information, characteristics of algorithm, program, analyzing programs. Arrays and Hash Tables Concept of sequential organization, linear and non-linear data structure, storage representation, array processing sparse matrices, transpose of sparse matrices.	**8**
3	**Stack, Queue, Linked List, and Python Implementations** Stack: concept, primitive operations, Stack Abstract Data Type, representation of stacks using sequential organization, basic stack operations. Applications of stack – introduction to expression evaluation and conversion (prefix, postfix), simple example of postfix evaluation. Queue: concept, representation of queue using array, introduction to circular queue and its basic implementation. Linked List: concept of linked organization, representation of linked list in memory. Singly linked list: basic operations – creation, traversing, searching, insertion, deletion. Doubly and circular linked list – overview and applications. Representation of stack and queue using linked list – introduction only. Practical examples in Python: traversing and searching in lists, stack push/pop using list,	**8**

	queue enqueue/dequeue using collections.deque.	
4	**Trees and Algorithms** Basic Terminology of Trees: Basic terminology of Trees Binary trees and its representation in memory. Algorithm Design Techniques, Performance Analysis of Algorithms, Types of Algorithm Analysis, Order of Growth, Asymptotic Notations, Recursion, Recurrences Relation, Substitution Method, Iterative Method, Recursion Tree, Master Theorem.	**8**
5	**Types of Algorithms** Divide and Conquer- Strassen's Matrix Multiplication, Backtracking- Backtracking Concept, N– Queens Problem, Sum of Subsets Problem, Branch and Bound: Introduction, Travelling Salesperson Problem, Greedy Algorithms- Optimal Merge Patterns, Huffman Coding, Knapsack Problem, Dynamic Programming- matrix multiplication.	**8**

Text Books

Sr. No.	Title	Author	Publisher
1	Introduction to Algorithms	T. Cormen	PHI Publication, 2nd Edition, 2002.

Reference Books

Sr. No.	Title	Author	Publisher
1	Data Structure and Algorithms	Aho, Ullman	Addison-Wesley Publication, 1st Edition,1983.
2	Algorithm Design – Foundation, Analysis & Internet Examples	Michel Goodrich, Roberto Tamassia	Wiley Publication, 2nd Edition,2006.
3	Algorithms in a Nutshell, A Practical Guide	George T. Heineman, Gary Pollice, Stanley Selkow	O'Reilly Media, 2nd Edition, 2016.
4	Fundamentals of Computer Algorithms	Ellise Horowitz, SartajSahni, S. Rajasekaran,	University Press (India) Private Ltd, 2nd Edition, 2008.
5	Computer algorithms: Introduction to Design and Analysis	Sara Base	Addison-Wesley Publication, 2nd Edition, 1988

Title of the Course: Database Management System Course Code: 25MCAC1103	L	T	P	Credit
	3	-	-	3

Course Prerequisite: A prior familiarity with computer organization, simple programming constructs, and basic concepts of data storage is recommended.

Course Description:
This course introduces the principles and practices of database systems. It emphasizes relational models, database design, and normalization techniques. Students will gain hands-on experience with SQL for creating, querying, and managing databases. Topics include transaction processing, concurrency control, recovery mechanisms, indexing, query optimization, and an overview of modern database trends such as NoSQL. By the end of the course, learners will be equipped to design, implement, and maintain reliable and efficient databases for real-world applications.

Course Objectives:
By the end of this course, students will be able to:
1. Understand core concepts of database systems and relational models.
2. Design well-structured databases using ER modeling and normalization.
3. Apply SQL commands to create, manipulate, and query databases effectively.
4. Demonstrate knowledge of transaction management, concurrency, and recovery techniques.
5. Evaluate database performance using indexing, optimization, and exposure to NoSQL systems.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand the purpose and applications of database systems and explain fundamental concepts like ER modeling, keys, and database design.
CO2	Apply relational algebra and relational calculus to query relational databases.
CO3	Construct and execute SQL queries, including nested queries, joins, views, and transactions.
CO4	Analyze and apply normalization techniques to design efficient relational schemas.
CO5	Explain file organization and indexing methods such as B+ trees and B-trees.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3		2		2							
CO2	3	3	3	2	3							
CO3	3	3	3		3							
CO4	2	3	2	3	3							
CO5	3											

Assessment Scheme:
Two components of Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and **CA2** are based on Surprise test/ Assignment/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content.
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	**Introduction** Database System Applications, Purpose of Database Systems, View of Data, Relational Databases, Database Design, Data Storage and Querying, ER model concepts, notation for ER diagram, Constraints, keys, E-R Diagrams, Mapping Cardinality, Concepts of Super Key, candidate key, primary key, weak entity sets, Cod's rules, Extended ER model, Generalization, Aggregation, , Reduction of an ER diagrams to tables.	**8**
2	**Relational Algebra and SQL** Structure of Relational Databases, Database Schema, Keys Relational algebra: Fundamental Operations, Additional Relational Algebra Operations, Extended Relational Algebra Operations. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities.	**8**
3	**Introduction to SQL** Overview of SQL, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operators, Set Operations, Null Values, Aggregate Functions, Nested Sub queries, Modification of the Database Intermediate SQL: Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schema, Authorization, Advanced SQL: Assessing SQL from Programming Language Functions and Procedures, Triggers.	**8**
4	**Relational Database Design** Normalization: Features of good relational designs, Functional dependencies, Normal forms, First, Second, third normal forms, BCNF, Functional Dependency Theory, Multivalued Dependencies, Fourth Normal Form, File Organization, Ordered Indices, B+ trees Index Files, B tree Index File.	**8**

5	**Transaction Processing** Transaction Concept, a simple Transaction Model, Transaction Atomicity and Durability, Transaction Isolation, ACID Properties, Serializability Concurrency Control Techniques: Lock based Protocols, Deadlock handling, Time Stamp-Based Protocols.	**8**

Text Books

Sr. No.	Title	Author	Publisher
1	Database System Concepts	Henry Korth, Abraham Silberschatz& S. Sudarshan,	McGraw-Hill Education, 6th edition 2011.

Reference Books

Sr. No.	Title	Author	Publisher
1	Database Management Systems	Raghu Ramakrishna, Johannes Gehrke	McGraw-Hill Education, 3rd edition 2003.
2	Murach's Oracle SQL and PL/SQL for Developers	Joel Murach,	Mike Murach & Associates,2nd edition 2014
3	Database Design	Wiederhold	McGraw-Hill Education, 2nd edition 1983.
4	Fundamentals of Database System	Navathe	Addison-Wesley Publication, 6th edition, 2012
5	Fundamentals of Database Management System	Mark L. Gillenson	Wiley Publication,2nd edition,2011

Title of the Course: Probability and Statistics Course Code: 25MCA1104	L	T	P	Credit
	3	1	-	4

Course Prerequisite: Basic Set Theory and Logic, Introductory Probability, Understanding of Functions and Graphs, Basic Statistics.

Course Description:

This course provides a foundational understanding of probability theory and statistical methods essential for data analysis and decision-making. It covers various approaches to probability, including classical, empirical, and axiomatic definitions, along with theorems such as Bayes' and the laws of addition and multiplication. The course introduces random variables, probability distributions, and the concept of mathematical expectation, followed by in-depth study of theoretical distributions like binomial, Poisson, and normal. It also covers correlation, regression analysis, and the principles of sampling and estimation, including hypothesis testing. Emphasis is placed on both theoretical understanding and practical application of statistical techniques.

Course Objectives:

By the end of this course, the students will be able to:

1. To introduce the foundational concepts of probability theory, including classical, empirical, and axiomatic approaches.
2. To explain and apply the concepts of random variables, probability distributions, and mathematical expectation.
3. To understand and analyze theoretical probability distributions such as binomial, Poisson, and normal, and their real-world applications..
4. To develop the ability to measure and interpret correlation using Karl Pearson's and Spearman's methods.
5. To build skills in linear regression analysis, including the derivation and interpretation of regression lines and coefficients.
6. To enable students to perform statistical sampling, estimation, and hypothesis testing for making data-driven decisions.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand and apply the basic concepts and theorems of probability including classical, empirical, and axiomatic approaches, Bayes theorem, and conditional probability.
CO2	Explain and analyze random variables, probability distributions (discrete and continuous), and compute expectations, variances, and standard distributions (Binomial, Poisson, and Normal).
CO3	Analyze and compute correlation coefficients using Karl Pearson's and Spearman's methods, and interpret the strength and direction of correlation.
CO4	Understand and perform linear regression analysis, derive regression lines, and interpret regression coefficients.
CO5	Understand and apply the basic concepts and theorems of probability including classical, empirical, and axiomatic approaches, Bayes theorem, and conditional probability.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3		2									
CO2	3	3										
CO3	2	3	2	2								
CO4	3	3	2	2								
CO5	3	3	3	3	2							

Assessment Scheme:
Two components of Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and **CA2** are based on Surprise test/ Assignment/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content.
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	**Probability Theory** Definition of probability: classical, empirical and axiomatic approach of probability, Addition theorem of probability, Multiplication theorem of probability, Bayes theorem of inverse probability, Properties of probabilities with proofs, Examples. Conditional Probability, Non Conditional Probability, Karl Pearson, Coefficient- shortcut method, extensive method.	**8**
2	**Random Variable and Mathematical Expectation** Random variables, Probability distributions, Probability mass function, Probability density function, Mathematical expectation, Join and marginal probability distributions, Properties of expectation and variance with proofs. Theoretical Probability Distributions: Binomial distribution, Poisson distribution, Normal distribution, Fitting of binomial distributions, Properties of binomial, Poisson and normal distributions, Relation between binomial and normal distributions, Relation between Poisson and normal distributions, Importance of normal distribution, Examples.	**9**
	Correlation	

3	Knowledge in AI, Knowledge Representation importance in AI, Knowledge Representation techniques: Propositional Logic, First-Order Logic (FOL), Semantic Networks, Frames, Scripts, Rule-Based Systems, Reasoning in AI, Simple Types of Reasoning: Deductive, Inductive, Abductive, Rule-Based Systems, Semantic Networks (Simple Diagrams), Ontologies (Simple Definitions of Concepts)	**8**
4	**Linear Regression Analysis** Introduction, Linear and non-linear regression, Lines of regression, Derivation of regression lines of y on x and x on y, Angle between the regression lines, Coefficients of regression, Theorems on regression coefficient, Properties of regression coefficient.	**6**
5	**Sampling and Estimation** Large Sample Estimation of a Population Mean, Small Sample Estimation of a Population Mean, Large Sample Estimation of a Population Proportion, Sample Size Considerations, Testing. Sampling Techniques, Random, Stratified, Systematic Sampling Distributions, Central Limit Theorem, Estimation Point, Estimation Interval, Estimation Confidence, Intervals for Mean and Proportion Hypotheses, The Elements of Hypothesis Testing, Large Sample Tests for a Population Mean, The Observed Significance of a Test. Small Sample Tests for a Population Mean, Large Sample Tests for a Population Proportion.	**9**

Text Books

Sr. No.	Title	Author	Publisher
1	Fundamentals of Statistics	Fundamentals of Statistics	Fundamentals of Statistics

Reference Books

Sr. No.	Title	Author	Publisher
1	Probability and Random Processes	G. V. Kumbhojkar	C. Jamnadas and Co., 14th Edition,2010
2	Advanced Engineering Mathematics	Erwin Kreyszig	John Wiley & Sons ,2006
3	Probability, Queuing Theory and Reliability Engineering	G. Haribaskaran	Laxmi Publication,2nd edition 2009.
4	Probability and Statistics	Murray Spiegel, John Schiller, R. ALU Srinivasan	Schaum's Outlines,4th edition,2013
5	Probability, Statistics with Reliability, Queuing and Computer Science Applications	Kishor S. Trivedi	WileyIndia Pvt.Ltd.,2nd edition,2011

Title of the Course: Java Programming Course Code: 25MCA1105	L	T	P	Credit
	3	0	-	3

Course Prerequisite: Students should have a basic understanding of OOP Concept, OS Basics, Command line & IDE

Course Description:

This course provides a comprehensive introduction to Java programming, focusing on both procedural and object-oriented programming concepts. It covers Java fundamentals such as data types, operators, control flow, arrays, and string handling. Students will learn to create and use classes, objects, constructors, inheritance, interfaces, and packages. The course also introduces multithreading, exception handling, and inter-thread communication. Additionally, it covers GUI development using AWT and Swing, event handling, and applet programming for interactive applications. Emphasis is placed on writing efficient, reusable, and user-friendly Java applications.

Course Objectives:

By the end of this course, students will be able to:

1. To introduce the fundamental concepts of Java programming, including data types, control structures, arrays, and strings.
2. To develop proficiency in object-oriented programming using Java through classes, objects, inheritance, and interfaces.
3. To enable students to build programs using constructors, static members, and inner classes for better code structure and reuse.
4. To provide an understanding of multithreading, thread lifecycle, synchronization, and exception handling in Java.
5. To familiarize students with the development of graphical user interfaces (GUI) using AWT and Swing components.
6. To help students design and implement event-driven applications and simple applets using Java's event-handling mechanisms.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand the features of Java, the Java programming environment, and apply fundamental programming constructs including data types, variables, operators, strings, control flow, and arrays.
CO2	Design and implement Java classes and objects, including constructors, static members, inner and anonymous classes, interfaces, inheritance, and packages.
CO3	Demonstrate knowledge of multithreading concepts including thread life cycle, synchronization, inter-thread communication, and exception handling in Java.
CO4	Develop simple applet-based applications, understand applet life cycle, and implement basic AWT components with event handling.
CO5	Design and implement GUI applications using AWT and Swing components, including layout managers and event handling mechanisms, understanding the differences between AWT and Swing.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3					2						
CO2	3		3		2							
CO3	3		3		3							
CO4	3	2	3		3		2					
CO5	3	2	3	2	3		2					

Assessment Scheme:
Two components of Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%, 30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and **CA2** are based on Surprise test/ Assignment/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content.
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	**Introduction** Features of Java, the Java Programming environment – (JDK, command line tools), Fundamental Programming structures in Java – (data types, variables, operators, strings, input and output control flow and arrays).	**8**
2	**OOPS Concepts** Class, Object, using predefined classes, defining your own classes, constructors, static data member and methods, inner classes and anonymous classes, introduction to interface, its structure and implementation Inheriting variables and methods in class, inheritance and constructors, abstract class and final class, object wrapper and auto boxing, inheritance and interfaces, introduction to packages.	**8**
3	**Threads & Exceptional Handling** Introduction, life cycle of a thread, thread states, thread properties, methods in Threads and Runnable, setting priority of threads, synchronization and inter thread communication, introduction to exception handling, predefined and user defined exceptions.	**8**

4	**Applet & AWT** Introduction to applet, life cycle of applet, development and execution of simple applet, drawing simple geometry shapes in applet, Introduction to AWT, events, listeners, event handling methods, a small application to demonstrate use of controls – label, button, check box, text, radio button, layout.	**8**
5	**Swing & Event Handling** GUI (Graphical User Interface), AWT (Abstract Window Toolkit), Swing (Advanced Components, Containers, Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Event Handling, Event Delegation Model, Event Sources and Listeners, Common Event Classes, Common Listener, Interfaces Swing Components (javax.swing), Event Handling, Introduction to swing, difference between swing and AWT in Swing.	**9**

Text Books

Sr. No.	Title	Author	Publisher
1	Java Programming	E. Balagurusamy	TMH.
2	The Complete Reference -Java 2	Schildt, Herbert	TMH

Reference Books

Sr. No.	Title	Author	Publisher
1	Core Java – Volume 1 Fundamentals	Cay S. Horstmann Gray Cornell	C. Jamnadas and Co., 8th Edition, Pearson Education
2	Programming with Java	E. Balagurusamy	3rd edition TMH ,2007
3	Java Server Programming Java EE 7 (J2EE 1.7) -Black Book	Kogent, Dreamtech Press Senn, LA.,	Laxmi Publication,2nd edition 2009.

Title of the Course: Data Structure and Algorithms using Python Laboratory Course Code: 25MCA1102L	L	T	P	Credit
	-	-	2	1

Course Prerequisite: Basic knowledge of programming logic and algorithm design (e.g., flowcharts or pseudocode), Understanding of computer fundamentals (data types, memory, input/output), exposure to any programming language (like C/C++/Java).

Course Description:

This course introduces students to the Python programming language for solving computational problems across a range of application domains. Beginning with foundational programming constructs like variables, control flow, and data structures, the course progresses to more advanced topics such as functions, object-oriented programming, and error handling. A strong emphasis is placed on writing clear, efficient, and maintainable code. Additionally, the course integrates the use of Python's powerful standard and third-party libraries for data analysis, scientific computing, and real-time applications.

Course Objectives:

By the end of this course, the students will be able to:

1. Understand the fundamental syntax and semantics of Python.
2. Apply conditional and iterative constructs to solve computational problems.
3. Implement classic searching and sorting algorithms using Python.
4. Work with Python's built-in data structures like lists, tuples, sets, and dictionaries.
5. Develop modular programs using functions, modules, and libraries.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Apply basic Python syntax and expressions to write simple programs involving variable manipulation and mathematical operations.
CO2	Develop solutions to scientific and logical problems using conditionals and loops.
CO3	Implement search and sorting algorithms using Python.
CO4	Manipulate and process data using built-in data structures (Lists, Tuples, Sets, Dictionaries).
CO5	Design and implement modular programs using functions, strings, and user-defined modules.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3				2							
CO2	3	3	3	2	3							
CO3	3	3	2		3							
CO4	3	3	2		3							
CO5	3	2	3	2	3							

Assessment Scheme:
Two components of in semester evaluation (CA1 and CA2) having 50% weightage of course contents and POE is having 50% weightage of total course contents.

Assessment Component	Marks
CA1	25
CA2	25
POE	50

Course Contents

Unit No.	Unit Title and Contents	Hours
1	Python programming using simple statements and expressions (exchange the values of two variables, circulate the values of n variables, distance between two points).	**2**
2	Scientific problems using Conditionals and Iterative loops.	**2**
3	Linear search and Binary search.	**2**
4	Selection sort, Insertion sort.	**2**
5	Merge sort, Quick Sort.	**2**
6	Implementing applications using Lists, Tuples.	**2**
7	Implementing applications using Sets, Dictionaries.	**2**
8	Implementing programs using Functions.	**2**
9	Implementing programs using Strings.	**2**
10	Implementing programs using written modules and Python Standard Libraries (pandas, numpy, Matplotlib, scipy)	**2**
11	Implementing real-time/technical applications using Exception handling.	**2**
12	Creating and Instantiating classes.	**2**

Text Books			
Sr. No.	**Title**	**Author**	**Publisher**
1	Python Programming: An Introduction to Computer Science	John Zelle	3rd Edition Franklin, Beedle & Associates.

Reference Books			
Sr. No.	**Title**	**Author**	**Publisher**
1	Learning with Python: How to Think Like a Computer Scientist	Allen B. Downey, Jeffrey Elkner, Chris Meyers	Green Tea Press
2	Python for Data Analysis	Wes McKinney	O'Reilly Media

Title of the Course: Database Management System Lab Course Code: 25MCAC1103L	L	T	P	Credit
	-	-	2	1

Course Prerequisite: Basic knowledge of relational database concepts,Understanding of data types, normalization, and ER modeling, Familiarity with basic programming logic (control structures, loops, functions).

Course Description:

This laboratory course introduces students to hands-on experience with relational database systems using SQL and PL/SQL. It emphasizes creating and managing databases, designing queries with constraints, joins, subqueries, group functions, and views. The course also covers stored procedures, cursors, and triggers. In addition, students gain exposure to NoSQL databases such as MongoDB, enabling them to perform CRUD operations and design small- scale real-world applications like office or hospital management systems.

Course Objectives:

By the end of this course, students will be able to:

1. To provide practical experience in SQL and PL/SQL programming.
2. To understand data manipulation, constraints, stored procedures, cursors, triggers, and joins.
3. To develop hands-on skills in database design and NoSQL database systems like MongoDB.
4. To implement real-world database applications.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Use SQL commands (DDL, DML, DCL) to create and manage databases.
CO2	Apply constraints, nested queries, and group functions for advanced data handling.
CO3	Develop and execute stored procedures, cursors, and triggers using PL/SQL.
CO4	Design and query relational databases using joins, views, and aggregations.
CO5	Use NoSQL tools like MongoDB to perform CRUD operations.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2										
CO2	3	3	2	2								
CO3	3		3	3	2							
CO4	3	3	3	3	3							
CO5	2		3		3							

Assessment Scheme:

Two components of in semester evaluation (CA1 and CA2) having 50% weightage of course contents and POE is having 50% weightage of total course contents.

Assessment Component	Marks
CA1	25
CA2	25
POE	50

Course Contents

Unit No.	Unit Title and Contents	Hours
1	Creating database tables and using data types.	**2**
2	Create table, modify table, Drop table.	**2**
3	Practical Based on Data Manipulation. Adding/Modify/Delete data using Insert/ Update/ Delete.	**2**
4	Practical Based on Implementing the Constraints. NULL and NOT NULL, Primary Key Constraint, Foreign Key Constraint Unique Constraint, Check Constraint, Default Constraint.	**2**
5	Practical for Retrieving Data Using following clauses. Simple select clause Accessing specific data with Where Clause Ordered By/ Distinct/Group By Clause.	**2**
6	Practical Based on Aggregate Functions. AVG, COUNT, MAX, MIN, SUM, CUBE	**2**
7	Practical Based on implementing Date and Time Functions.	**2**
8	Practical Based on implementing use of UNION, INTERSECTION, SET DIFFERENCE.	**2**
9	Implement Nested Queries & all types of JOIN operation	**2**
10	Practical Based on implementing use of Triggers.	**2**
11	Practical Based on implementing Cursor.	**2**

Text Books			
Sr. No.	**Title**	**Author**	**Publisher**
1	Database System Concepts	Henry Korth, Abraham Silberschatz& S. Sudarshan	McGraw-Hill Publication, 6th Edition, 2011**.**
2	Agile Software Development, Principles, Patterns and Practices	Robert C. Martin	Prentice Hall.

Reference Books			
Sr. No.	**Title**	**Author**	**Publisher**
1	Database Management Systems	Raghu Ramakrishna, Johannes Gehrke	McGraw- Hill Publication, 3rd Edition, 2003
2	Murach's Oracle SQL and PL/SQL for Developers	Joel Murach	Mike Murach & Associates, 2nd Edition, 2014
3	Database Design	Wiederhold	McGraw-Hill Publication, 2nd Edition, 1983
4	Fundamentals of Database System	Navathe	Addison-Wesley Publication, 6th Edition, 2012.
5	Fundamentals of Database Management System	Mark L. Gillenson	Wiley Publication, 2ndEdition, 2011.

Title of the Course: Java Programming Laboratory Course Code: 25MCA1105L	L	T	P	Credit
	-	-	2	1

Course Prerequisite: OOP Concept, Java Basics.

Course Description:

This lab course provides hands-on experience in core Java programming, including classes, interfaces, packages, and exception handling. Students will learn to implement multithreading, develop GUI applications using AWT, and create applets. The course also covers file handling, JDBC for database connectivity, and basic networking with socket programming. It emphasizes practical skills to build real-world Java applications using object-oriented principles.

Course Objectives:

By the end of this course, the students will be able to:

1. To develop a strong understanding of object-oriented programming concepts using Java.
2. To provide practical experience in creating Java classes, packages, interfaces, and handling exceptions.
3. To enable students to implement multithreading and thread communication in Java applications.
4. To design and build GUI-based applications using AWT and applets.
5. To perform file operations and manage data using Java I/O streams.
6. To connect Java applications to databases using JDBC and understand basic client-server communication through networking.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Apply object-oriented programming principles to develop Java programs using classes, objects, inheritance, polymorphism, and encapsulation.
CO2	Construct Java applications by implementing packages, interfaces, and exception handling mechanisms for modular and robust programming.
CO3	Develop multithreaded Java programs with proper synchronization and inter-thread communication.
CO4	Design and implement graphical user interfaces (GUIs) using AWT components and applets.
CO5	Perform file handling operations using Java I/O streams to store, retrieve, and manipulate data and integrate java applications with database.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2	3	3	3		3							
CO3	3		3									
CO4	3	3	3		3							
CO5	3	2	3	2	2							

Assessment Scheme:
Two components of in semester evaluation (CA1 and CA2) having 50% weightage of course contents and POE is having 50% weightage of total course contents.

Assessment Component	Marks
CA1	25
CA2	25
POE	50

Course Contents

Unit No.	Unit Title and Contents	Hours
1	Demonstration of installation and configuration of Jdk with Command line. a) Write a java program to take user input and print it. b) Write a java program to swap two numbers.	**2**
2	Write a java program to find the largest number out of n natural numbers.	**2**
3	. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.	**2**
4	Write a Java program to print the first N numbers in the Fibonacci series using a for loop.	**2**
5	Write a Java program to demonstrate single inheritance by creating a base class Animal with a method eat() and a derived class Dog that inherits from Animal and has its own method bark().	**2**
6	Write a Java program to demonstrate method overloading by creating a class MathOperations with overloaded methods add(int, int), add(double, double), and add(int, int, int).	**2**
7	Write a Java program to create an interface Playable with a method play(). Create classes Football and Basketball that implement the interface. Use polymorphism to call the play() method.	**2**
8	Write a Java program to create an abstract class Shape with a field color and an abstract method calculateArea(). Create subclasses Circle and Rectangle that provide concrete implementations for the abstract method.	**2**
9	Write a Java program to create a package named shapes and define a class Circle within it. The class should include methods to calculate the area and circumference of a circle. Create another class in a different package that uses the Circle class.	**2**

10	Write a Java program to create a thread by extending the Thread class and overriding its run() method. Start the thread and print a message from within the run () method.	**2**
11	Write a Java program to create a thread by implementing the Runnable interface. Implement the run () method and use a Thread object to execute the Runnable.	**2**
12	Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.	**2**
13	Write a program to create a frame using AWT. Implement mouseClicked(), mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it.	**2**
14	Using AWT, write a program to create two buttons named "Red" and "Blue". When a button is pressed the background colour should be set to the colour named by the button's label.	**2**
15	Java Program to Display Image using Applet.	**2**

Text Books

Sr. No.	Title	Author	Publisher
1	Java Programming	E. Balagurusamy	TMH.
2	The Complete Reference -Java 2	Schildt, Herbert	TMH

Reference Books

Sr. No.	Title	Author	Publisher
1	Core Java – Volume 1 Fundamentals	Cay S. Horstmann Gray Cornell	C. Jamnadas and Co., 8th Edition, Pearson Education
2	Programming with Java	E. Balagurusamy	3rd edition TMH ,2007
3	Java Server Programming Java EE 7 (J2EE 1.7) -Black Book	Kogent, Dreamtech Press Senn, LA.,	Laxmi Publication,2nd edition 2009.

Title of the Course: Web Technology Laboratory Course Code: 25MCA1106L	L	T	P	Credit
	-	-	2	1

Course Prerequisite: Basic knowledge of computer operations and internet browsing.**.** Familiarity with programming concepts (variables, conditions, loops) is helpful but not mandatory, understanding of basic markup language (HTML).

Course Description:

This course introduces the fundamental concepts of web design and development, covering both the front-end and back-end technologies essential for creating dynamic and interactive websites. Students will learn HTML, CSS, JavaScript, PHP, JQuery, AJAX, and MySQL integration. The course emphasizes hands-on experience in web page layout, styling, scripting, form handling, and database connectivity to develop robust web applications.

Course Objectives:

By the end of this course, the students will be able to:

1. Understand the principles and process of website design and development.
2. Utilize HTML and CSS to create well-structured, styled, and responsive web pages.
3. Apply JavaScript fundamentals to add interactivity and dynamic behavior to web pages.
4. Use PHP to handle server-side programming, form processing, and session management.
5. Integrate JQuery and AJAX to enhance user experience with client-server communication.
6. Connect web applications to MySQL databases for storing and managing data.Interpret results and write research papers using proper tools and ethics.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand the fundamentals of web design, HTML structure, and web development process including hyperlinks, tables, forms, and images.
CO2	Apply Cascading Style Sheets (CSS) to design visually appealing and well-formatted web pages, including styling tables and importing data.
CO3	Develop dynamic web pages using JavaScript programming concepts such as variables, control structures, functions, and event handling.
CO4	Implement server-side programming with PHP to handle form data, sessions, and integrate with HTML forms effectively.
CO5	Use JQuery for form validation and UI enhancements, and employ AJAX techniques along with PHP for asynchronous web applications.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2			2							
CO2	3	3	3		3	2						
CO3	3	3	3		3	2	2					
CO4	3	3	3	2	3	3	3	2				
CO5	3	3	3	2	3							

Assessment Scheme:

Two components of in Semester Evaluation (CA1 and CA2), having 50% weightage respectively.

Assessment Component	Marks
CA1	25
CA2	25
POE	--

CA1 and CA2 are based on practical performance.

Course Contents

Unit No.	Unit Title and Contents	Hours
1	Write an HTML code to display your CV on a web page	**2**
2	Design the following static web pages required for an online book store web site a) HOME PAGE: The static home page must contain three frames. b) LOGIN PAGE. c) CATOLOGUE PAGE: The catalogue page should contain the details of all the books available in the web site in a table. d) REGISTRATION PAGE	**2**
3	Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color,bold and font size).	**2**
4	Write JavaScript to validate the following fields of the Registration page. a) First Name (Name should contain alphabets and the length should not be less than 6 characters). b) Password (Password should not be less than 6 characters length). c) E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com) d) Mobile Number (Phone number should contain 10 digits only) e) Last Name and Address (should not be Empty).	**2**
5	Write a JavaScript to validate the following fields of employee on html form: email, name, mobile no., address, salary.	**2**
6	Write a program to design a simple calculator using: a) JavaScript b) PHP c) Servlet d) JSP	**2**
7	Write a PHP program to display a digital clock which displays the current time of the server.	**2**

8	Write a PHP program to sort the student records which are stored in the database using ascending/descending order.	**2**
9	Develop and demonstrate PHP Script for the following problems: a) Write a PHP Script to find out the Sum of the Individual Digits. b) Write a PHP Script to check whether the given number is Palindrome or not.	**2**
10	Write a PHP Program to display current Date, Time and Day.	**2**
11	Create an XML document that contains 10 users' information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.	**2**
12	Modify the above PHP program to use an xml instead of database.	**2**

Text Books

Sr. No.	Title	Author	Publisher
1	HTML 5 Black Book	DT Editorial Services	2nd edition (English, Paperback).

Reference Books

Sr. No.	Title	Author	Publisher
1	Learning PHP, MySQL & JavaScript with j Query, CSS & HTML5 Paperback	Robin Nixon	O'Reilly Publications.
2	Head First HTML & CSS	E. Robson, E. Freeman	O 'Reilly Media, 2nd Edition, 2012.

Semester II

Title of the Course: Object Design & Agile Development Course Code: 25MCA1201	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Basic programming knowledge. Understanding of software development lifecycle. Familiarity with object-oriented programming concepts

Course Description:

This course covers advanced software development methodologies, focusing on object-oriented programming and design using UML, distributed computing, and agile software development practices. It explores frameworks like Scrum and Kanban, design principles, and emphasizes software reuse, testing, and continuous integration for efficient and flexible software delivery.

Course Objectives:

1. Understand object-oriented concepts, design, and analysis using UML
2. Learn distributed object computing and middleware standards
3. Grasp agile methodologies, principles, and benefits
4. Apply Scrum and Kanban frameworks in project management

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Apply object-oriented concepts and modeling techniques using UML for software analysis and design.
CO2	Understand and evaluate distributed object computing and middleware standards such as COM, DCOM, and CORBA.
CO3	Explain the principles, values, and benefits of Agile software development methodologies.
CO4	Apply Scrum framework elements (roles, events, artifacts) to manage Agile software projects.
CO5	Apply Kanban framework and Agile testing practices including test-driven development and test automation.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P011
CO1	3	3	2	2	2	-	2	1	2	1	2	2
CO2	3	3	3	2	2	2	2	2	3	3	2	2
CO3	2	2	3	2	3	1	2	2	3	1	2	3
CO4	1	2	2	1	1	1	2	3	3	2	3	3
CO5	1	2	2	1	2	-	2	2	3	2	3	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	Review of programming practices and code-reuse; Object model and object-oriented concepts; Object-oriented programming languages and implementation. Object-oriented analyses and design using UML structural, behavioral and architectural modeling. Unified development process, Software reuse design patterns, components and framework;	**02**
2.	Distributed object computing, interoperability and middle ware standards COM/DCOM and CORBA; Object-oriented database system data model, object definition and query language, object relational system.	**02**
3.	Introduction Need of Agile software development, History of Agile, Agile context–manifesto, principles, methods, values. The benefits of agile in software development. Agile Design Methodologies Fundamentals, Design principles–Single responsibility, Open-closed, Liskov- substitution, Dependency inversion, Interface-segregation.	**02**
4.	Scrum Introduction to scrum framework, Roles: Product owner, team members and scrum master, Events: Sprint, sprint planning, daily scrum, sprint review, and sprint retrospective, Artifacts: Product backlog, sprint backlog and increments. User stories-characteristics and contents.	**02**
5.	Kanban Introduction to Kanban framework, Workflow, Limit the amount of work in progress, pulling work from column to column, Kanban board, adding policies to the board, The Agile lifecycle and its impact on testing, Test driven development–Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.	**02**

Text Books			
Sr. No.	**Title**	**Author**	**Edition**
1	Object Oriented System Analysis	Sally Shlaer	Prentice Hall PTR

2	Object Oriented System Analysis and Design using UML	Simon Bennett	McGraw-Hill.
3	Agile Software Development with Scrum	Ken Schawber, Mike Beedle	Pearson
4	Agile Software Development, Principles, Patterns and Practices	Robert C. Martin	Prentice Hall

Reference Books:

Sr. No.	Title	Author	Edition
1.	User Stories Applied: For Agile Software Development	Mike Cohn	Addison Wesley Signature Series
2.	Agile Testing: A Practical Guide for Testers and Agile Teams	Lisa Crispin, Janet Gregory	Addison Wesley
3.	Agile: The Complete Overview of Agile Principles and Practices	Paul VII	Agile Product Management
4.	Agile Software Development, Principles, Patterns, and Practices	Robert Martin	Pearson New International Edition

Title of the Course: Computer Networks **Course Code: 25MCA1202**	**L**	**T**	**P**	**Credit**
	3	1		**4**

Course Pre-Requisite:

Students should have a basic understanding of operating systems, computer architecture, and programming fundamentals in C or Java. Familiarity with binary arithmetic, logical operations, and basic hardware concepts is expected.

Course Description:

This course provides a comprehensive introduction to computer networking principles, architectures, protocols, and technologies. It covers layered network models, data transmission fundamentals, LAN/WAN technologies, protocols at each OSI layer, routing algorithms, and network security. The course also explores wireless networking and application layer protocols such as HTTP, DNS, and FTP. Real-world protocols like TCP/IP are emphasized, enabling students to understand and evaluate modern network systems.

Course Objectives:

By the end of this course, students will be able to:

1. Understand the architecture and protocols of computer networks.
2. Analyse and compare LAN technologies and wireless networks.
3. Apply error detection, correction, and flow control techniques in data link layer.
4. Understand routing algorithms, IP addressing, and congestion control methods.
5. Use application layer protocols and basic network security principles.
6. Develop problem-solving skills relevant to network design and performance evaluation.

Course Outcomes:

CO	**After the completion of the course the student should be able to**
CO1	Explain the basic concepts, applications, and models of computer networks including OSI and TCP/IP architecture.
CO2	Describe and differentiate various LAN and wireless technologies including Ethernet, Wi-Fi, Bluetooth, and WiMAX.
CO3	Explain the basic concepts, applications, and models of computer networks including OSI and TCP/IP architecture.
CO4	Describe and differentiate various LAN and wireless technologies including Ethernet, Wi-Fi, Bluetooth, and WiMAX.
CO5	Understand application layer protocols and the basics of network security including authentication and cryptography.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	2	2	2		2		1	1		2
CO2	3	2	2	2	3		2		1	1	1	2
CO3	3	3	3	2	3		2				1	3
CO4	3	3	3	3	3		2		1	1	1	2
CO5	3	2	2	2	3	2	2		2	3	1	2

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	**Introduction** Applications of computer networks, Network hardware, Network software: Protocol Hierarchy, Design Issue, connection oriented vs. connectionless, Service Primitives, Reference models: OSI and TCP/IP, Example networks: Internet, Network standardization, Performance: Bandwidth and Latency, Delay and bandwidth product, High- Speed Network, Application Performance Needs.	**08**
2.	**LAN Technologies** X5, Frame relay, ATM, Ethernet (802.3), FDDI, Token Rings, Resilient Packet Rings, Wireless LANs: Wi-Fi (802.11), Cell Phone Technologies, Broadband Wireless: Wi- MAX (802.16), Bluetooth (802.15.1), RFID.	**08**

3.	**Data Link Layer** Introduction, functions. Design Issues: Services to Network Layer, Framing. ARQ strategies: Error detection and correction, Parity Bits, Hamming Codes (11/12-bits) and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol. WAN Connectivity: PPP and HDLC. MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA, CSMA/CD, CSMA/CA, Binary Exponential Back-off algorithm, Introduction to Ethernet IEEE 802.3, IEEE 802.11 a/b/g/n, IEEE 802.15 and IEEE 802.16 Standards.	**08**

4	**Network Layer and Congestion Control** IPv4/IPv6, Routers and Routing Algorithms distance vector link state. TCP UDP and sockets, General principles, Congestion prevention policies, Load shading, Jitter control, Quality of service: Packet scheduling, Traffic shaping, integrated Services, **Routing Protocols**: RIP, OSPF, BGP, MPLS. **Routing in MANET**: AODV, DSR, Mobile IP.	**08**
5.	**Application Layer Protocols** DNS, SMTP, POP, FTP, HTTP. Network Security: Authentication, Basics of public key and private key cryptography, digital signatures and certificates, firewalls.	**07**

Text Books

Sr.No.	**Title**	**Author**	**Edition**
1	Computer Networks	A. Tanenbaum	5th Edition, 2011

Reference Books:

Sr. No.	**Title**	**Author**	**Edition**
1.	Data Communications and Networking	B. Forouzan	5th Edition, 2013
2.	Computer Networks: A Systems Approach	Larry Peterson and Bruce Davie	Morgan Kufman Publication, 5th Edition, 2012.
3.	An Engineering Approach to Computer Networking	S. Keshav	Addison-Wesley Professional
4.	Computer Networks and Internet	D. Comer	Pearson Education, 6th Edition, 2014
5.	Computer Communications and Networking Technologies	M. Gallo, W. Hancock	Brooks/Cole Publisher, 2001

Title of the Course: Artificial Intelligence **Course Code: 25MCA1203**	**L**	**T**	**P**	**Credit**
	3			**3**

Course Pre-Requisite:
Basic knowledge of Data Structures, Algorithms, Discrete Mathematics, and Programming fundamentals.

Course Description:
This course introduces the fundamentals of Artificial Intelligence, focusing on intelligent agents, search strategies, knowledge representation, logical reasoning, learning methods, and expert systems. It explores both theoretical foundations and practical applications in problem- solving and decision-making environments.

Course Objectives:

1. To understand the foundations and evolution of Artificial Intelligence.
2. To apply various search techniques for problem-solving.
3. To explore knowledge representation and reasoning mechanisms.
4. To implement learning models including decision trees and reinforcement learning.
5. To study expert systems and knowledge engineering principles.

Course Outcomes:

CO	**After the completion of the course the student should be able to**
CO1	Understand the fundamental concepts, history, and applications of Artificial Intelligence.
CO2	Apply problem-solving techniques including search algorithms, heuristics, and optimization methods.
CO3	Use knowledge representation methods such as logic, semantic networks, and frames.
CO4	Develop AI systems using reasoning, planning, and learning techniques including machine learning basics.
CO5	Understand natural language processing and computer vision fundamentals.

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	2	2	2		2		1	1		2
CO2	3	3	3	2	3		2		2	2	1	2
CO3	3	3	3	2	3		2		2	2	1	2
CO4	3	3	3	3	3		2		2	2	2	3
CO5	3	2	2	2	2		2		1	1	1	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.

MSE is based on 50% of course content

ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.	**08**
2.	Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A*, AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.	**08**
3.	Knowledge representation issues, predicate logic- logic programming, semantic nets-frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Baye's probabilistic interferences and dempstershafer theory.	**08**
4	First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.	**08**
5.	Expert systems- Introduction, basic concepts, structure of expert systems, the human element in expert systems how expert systems works, problem areas addressed by expert systems, expert systems success factors, types of expert systems, expert systems and the internet interacts web, components of an expert system, Knowledge base, Inference Engine, User interface, types of reasoning, forward chaining, backward chaining, characteristics of expert system. knowledge engineering, scope of knowledge, difficulties, in knowledge acquisition methods of knowledge acquisition, machine learning, intelligent agents, selecting an appropriate knowledge acquisition method, societal impacts reasoning in artificial intelligence, inference with rules, with frames: model based reasoning, case based reasoning, explanation & meta knowledge inference with uncertainty representing uncertainty	**8**

Text Books			
Sr. No.	**Title**	**Author**	**Edition**
1.	Artificial Intelligence – A Modern Approach	S. Russel and P. Norvig	Second Edition, Pearson Education
2	Computational Intelligence: a logical approach	David Poole, Alan Mack worth, Randy Goebel	Oxford University Press
3	Artificial Intelligence: Structures and Strategies for complex problem solving	G. Luger	Fourth Edition, Pearson Education
4	Artificial Intelligence: A new Synthesis	J. Nilsson	Elsevier Publishers

Title of the Course: Program Elective-I A) Internet of Things Course Code: 25MCA1204	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Basic knowledge of computer networks, communication technologies, and programming fundamentals.

Course Description:

This course introduces the fundamentals of the Internet of Things (IoT), including architecture, smart devices, networking, data analytics, and security. It explores real-world IoT applications through case studies in smart cities, healthcare, and agriculture.

Course Objectives:

1. To understand the foundational concepts, architecture, and impact of IoT.
2. To explore the role of sensors, actuators, and smart objects in IoT systems.
3. To learn about IP-based communication and protocols for IoT networking.
4. To introduce data analytics, machine learning, and security for IoT applications.
5. To examine IoT implementation strategies through real-world case studies.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand the fundamentals, architecture, and challenges of IoT systems and networks.
CO2	Analyze and explain the role of sensors, actuators, and smart objects in IoT systems.
CO3	Explain the use of IP-based communication protocols and optimization techniques for IoT networks.
CO4	Apply knowledge of data analytics, edge computing, and security mechanisms to IoT applications.
CO5	Evaluate real-world IoT case studies and develop IoT-based solutions using appropriate technology.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	1		2		1		1			2
CO2	2	3	2	2	2	1	2		2	1		2
CO3	3	3	3	2	3	1	2	1	2	2	2	2
CO4	2	2	3	3	3	2	2	2	3	2	3	3
CO5	3	2	3	3	3	2	3	2	3	3	3	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.

MSE is based on 50% of course content

ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	**Introduction** What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack	**08**
2.	**Smart Objects** The "Things" in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies.	**08**
3.	**IP as the IoT Network Layer** IP as the IoT Network Layer, The Business Case for IP, the need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods	**08**
4.	**Data and Analytics for IoT** Data and Analytics for IoT, An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Securing IoT. Attacks on IoT Devices.	**08**
5.	**Case Studies** An IoT Strategy for Smarter Cities, Smart City IoT Architecture, Smart City Security Architecture, Smart City Use-Case Examples. An IoT Strategy for healthcare System. An IoT Strategy for smart agriculture System.	**08**

Text Books			
Sr.No.	**Title**	**Author**	**Edition**
1	IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things	David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry	1stEdition, Pearson Education
2	Internet of Things	Srinivasa K G	CENGAGE Leaning India, 2017

Reference Books:			
Sr. No.	**Title**	**Author**	**Edition**
1.	Internet of Things (A Hands-On-Approach)	Vijay Madisetti and Arshdeep Bahga	1stEdition, VPT, 2014. (ISBN: 978-8173719547)
2.	Internet of Things: Architecture and Design Principles	Raj Kamal	1st Edition, McGraw Hill Education, 2017 ISBN: 978-9352605224

Title of the Course: Program Elective-I B) Human Computer Interaction Course Code: 25MCA1204	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Basic knowledge of computer programming and software development, Familiarity with web technologies (HTML, CSS, basic JavaScript) or UI frameworks

Course Description:

This course introduces the fundamental concepts, principles, and practices of Human-Computer Interaction (HCI). It focuses on the design, evaluation, and implementation of user-centered interfaces that enhance user experience and usability. Students will explore topics such as interaction design, user interface (UI) development, usability engineering, cognitive models, and accessibility. The course also covers modern trends in HCI, including touch, gesture, voice-based interfaces, and human-centered AI.

Course Objectives:

1. To introduce the fundamental concepts and principles of Human-Computer Interaction**,** focusing on designing user-centered systems.
2. To understand human cognitive and perceptual capabilities that influence the design of effective and usable interfaces.
3. To learn about IP-based communication and protocols for IoT networking.
4. To explore various interaction styles and interface design paradigms**,** including graphical, voice, gesture**,** and touch-based interfaces.
5. To develop the ability to evaluate user interfaces through usability testing, heuristic evaluation, and other assessment techniques.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the fundamental principles and components of Human-Computer Interaction and user-centered design.
CO2	Analyze user needs and behaviors to inform the design of effective and intuitive user interfaces.
CO3	Design interactive prototypes using appropriate tools and techniques for various types of user interfaces.
CO4	Evaluate user interfaces for usability, accessibility, and performance using standard evaluation methods.
CO5	Apply cognitive and perceptual principles to improve user experience in software and web applications

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011
CO1	3	2	1		2		1		1		
CO2	2	3	2	2	2	1	2		2	1	
CO3	3	3	3	2	3	1	2	1	2	2	2
CO4	2	2	3	3	3	2	2	2	3	2	3
CO5	3	2	3	3	3	2	3	2	3	3	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.

MSE is based on 50% of course content

ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	**Fundamentals:** The Human: I/O channels – Memory – Reasoning and problem solving; The Computer: Devices– Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles –elements – interactivity- Paradigms. - Case Studies.	**08**
2.	**Design & Software Process:** Interactive Design: Basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process: Software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules: principles, standards, guidelines, rules. Evaluation Techniques – Universal Design	**08**
3.	**Models & Theories:** HCI Models: Cognitive models: Socio-Organizational issues and stakeholder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.	**08**
4.	**Mobile HCI:** Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools. - Case Studies	**08**
5.	**Web Interface Design:** Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow - Case Studies	**08**

Text Books			
Sr.No.	**Title**	**Author**	**Edition**
1	Human Computer Interaction	Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale	3rd Edition, Pearson Education, 200
2	Brian Fling	Mobile Design and Development	First Edition, O'Reilly Media Inc., 2009
3	Bill Scott and Theresa Neil	Designing Web Interfaces	First Edition, O'Reilly, 2009

Title of the Course: Program Elective-I C) Advance Database Technologies Course Code: 25MCA1204	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Strong understanding of Relational Database Management Systems (RDBMS), Basic knowledge of programming languages

Course Description:

This course provides an in-depth exploration of advanced concepts in database systems beyond the fundamentals of relational databases. It covers emerging database models and technologies such as NoSQL databases, distributed databases, object-oriented databases, data warehousing, data mining, and database security. Students will gain hands-on experience with modern database platforms and learn to handle large-scale, complex, and unstructured data. The course also introduces advanced topics like query optimization, transaction management, and big data technologies.

Course Objectives:

1. To explore advanced data models and database architectures, including NoSQL, distributed, and object-oriented databases.
2. To understand the principles of data warehousing and data mining for analytical processing and decision-making.
3. To study advanced transaction management, query optimization, and database performance tuning techniques.
4. To introduce database security, access control, and recovery techniques to ensure data integrity and confidentiality.
5. To provide practical exposure to modern database tools and technologies used in big data and enterprise applications.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Analyze and apply advanced data models such as NoSQL, object-oriented, and distributed databases for specific application needs.
CO2	Design and implement data warehousing solutions and perform basic data mining operations for business intelligence.
CO3	Optimize complex queries and manage database transactions effectively to improve system performance and reliability.
CO4	Evaluate and implement database security techniques, including access control, encryption, and backup/recovery mechanisms.
CO5	Use modern database technologies and tools to manage large-scale and unstructured data in real-world scenarios.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	1		2		1		1			3
CO2	2	3	2	2	2	1	2		2	1		3
CO3	3	3	3	2	3	1	2	1	2	2	2	2
CO4	2	2	3	3	3	2	2	2	3	2	3	2

CO5	3	2	3	3	3	2	3	2	3	3	3	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.

MSE is based on 50% of course content

ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	Distributed Systems – Introduction – Architecture – Distributed Database Concepts – Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing	**08**
2.	Active Databases Model – Design and Implementation Issues - Temporal Databases – Temporal Querying - Spatial Databases: Spatial Data Types, Spatial Operators and Queries – Spatial Indexing and Mining – Applications -– Mobile Databases: Location and Handoff Management, Mobile Transaction Models – Deductive Databases - Multimedia Databases.	**08**
3.	NoSQL – CAP Theorem – Sharding - Document based – MongoDB Operation: Insert, Update, Delete, Query, Indexing, Application, Replication, Sharding– Cassandra: Data Model, Key Space, Table Operations, CRUD Operations, CQL Types – HIVE: Data types, Database Operations, Partitioning – HiveQL – OrientDB Graph database – OrientDB Features	**08**
4.	Structured, Semi structured, and Unstructured Data – XML Hierarchical Data Model – XML Documents – Document Type Definition – XML Schema – XML Documents and Databases –XML Querying – XPath – XQuery	**08**
5.	IR concepts – Retrieval Models – Queries in IR system – Text Preprocessing – Inverted Indexing– Evaluation Measures – Web Search and Analytics – Current trends.	**08**

Text Books			
Sr.No.	**Title**	**Author**	**Edition**
1	Database System Concepts	Abraham Silberschatz, Henry F Korth, S. Sudharshan	Seventh Edition, McGraw Hill, 2019
2	Fundamentals of Database Systems	R. Elmasri, S.B. Navathe	Seventh Edition, Pearson Education/Addison Wesley, 2017
3	Next Generation Databases, NoSQL, NewSQL and Big Data	Guy Harrison	First Edition, Apress publishers, 2015

Reference Books			
Sr.No.	**Title**	**Author**	**Edition**
1	Data Mining: Concepts and Techniques	Jiawei Han, MichelineKamber, Jian Pe	Third Edition, Morgan Kaufmann, 2012
2	Teach Yourself NoSQL with MongoDB in 24 Hours	Brad Dayley	Sams Publishing, First Edition, 2014.
3	An Introduction to Database Systems	C. J. Date, A. Kannan, S. Swamynathan	Eighth Edition, Pearson Education, 2006

Title of the Course: Open Elective-I A)Research Methodology Course Code: 25MCA1205	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Basic understanding of statistics, data handling, academic writing, and foundational knowledge in business or social sciences.

Course Description:

This course provides an overview of research methodology, including research design, hypothesis formulation, data collection, sampling, analysis, and scientific writing. It emphasizes both qualitative and quantitative methods and introduces tools for academic research and publishing, particularly in computer science.

Course Objectives:

1. Understand the fundamentals and types of research.
2. Develop skills in formulating research problems and hypotheses.
3. Learn various research designs and measurement techniques.
4. Apply sampling methods and perform basic data analysis.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Understand foundational concepts and language of research including theory, hypothesis, variables, and scientific methods.
CO2	Apply research methodology and scientific process for formulating problems, hypotheses, and designing research.
CO3	Identify appropriate research designs and compare qualitative and quantitative research approaches.
CO4	Understand concepts of measurement and sampling; evaluate tools for data collection and analysis.
CO5	Develop skills in data interpretation, academic writing, ethical publishing, and use of research tools/databases.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	1	2	2	1	2	3	1	2	3	3
CO2	3	3	3	3	3	1	2	2	1	1	2	3
CO3	2	3	3	3	2	1	2	2	1	1	2	3
CO4	3	3	3	3	3	1	2	2	1	1	2	3
CO5	1	2	1	3	3	3	3	3	3	3	3	3

Assessment Scheme:
Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively..

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA1 and CA2 are based on Assignment/ Declared test/ Quiz/Seminar/Group discussions presentation, etc.
MSE is based on 50% of course content
ESE is based on 100% course content with 60-70% weightage for course content covered after MSE

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	Foundations of Research: Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Postulate, Proposition, Thesis, Hypothesis, Law, Principle. Research methods vs Methodology, Need of Research in Business and Social Sciences, Objectives of Research, Issues and Problems in Research, Characteristics of scientific method - Understanding the language of Research - Concept, Construct, Definition, Variable Research Process.	**08**
2.	Research Methodology: Meaning of Research Methodology, Stages in Scientific Research Process: Identification and Selection of Research Problem, Formulation of Research Problem, Review of Literature, Formulation of Hypothesis, Formulation of research Design, Sample Design, Data Collection, Data Analysis, Hypothesis testing and Interpretation of Data, Preparation of Research Report Problem Identification & Formulation - Research Question - Investigation Question - Measurement Issues - Hypothesis - Qualities of a good Hypothesis, Null Hypothesis & Alternative Hypothesis. Hypothesis Testing - Logic & Importance.	**08**
3.	Research Design: Concept and Importance in Research - Features of a good research design - Exploratory Research Design - concept, types and uses, Descriptive Research Designs - concept, types and uses. Experimental Design: Concept of Independent & Dependent variables. Qualitative and Quantitative Research: Qualitative research - Quantitative research - Concept of measurement, causality, generalization, replication. Merging the two approaches.	**08**
4.	Measurement: Concept of measurement- what is measured? Problems in measurement in research- Validity and Reliability. Levels of measurement, Nominal, Ordinal, Interval, Ratio. Sampling: Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non Response. Characteristics of a good sample. Probability Sample- Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample, Data Analysis: Data Preparation - Univariate analysis (frequency tables, bar charts, pie charts, percentages),	**08**

5.	Interpretation of Data and Paper Writing- Layout of a Research Paper, Journals in Computer Science, Impact factor of Journals, When and where to publish? Ethical issues related to publishing, Plagiarism and Self-Plagiarism. Use of Encyclopedias, Research Guides, Handbook etc., Academic Databases for Computer Science. Discipline. Use of tools I techniques for Research: methods to search required information effectively, Reference Management Software like Zotero/Mendeley, Software for paper formatting like LaTeX/ MS Office, Software for detection of Plagiarism	**08**

Text Books

Sr. No.	**Title**	**Author**	**Edition**
1	Fundaentals of Statistics	Fundamentals of Statistics	Fundamentals of Statistics
2	Research Methodology	C. R. Kothari	
3	Select references from the Internet	Google Scholar	-

Reference Books:

Sr. No.	**Title**	**Author**	**Publisher**
1.	Business Research Methods	Alan Bryman & Emma Bell	Oxford University Press.

Title of the Course: Open Elective-I B) Cyber Laws Course Code: 25MCA1205	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:
Basic knowledge of computers, internet, and general legal framework in India.

Course Description:
This course provides an overview of cyber laws, e-commerce regulations, and e-governance in India with reference to a free-market economy. It covers electronic records, digital signatures, intellectual property rights, international cyber law efforts, offences under the IT Act, and related legal provisions.

Course Objectives:

1. To understand the legal framework governing e-commerce, e-governance, and cyberspace in India.
2. To study laws on electronic records, digital signatures, and intellectual property in cyberspace.
3. To explore international initiatives and conventions on cybercrime.
4. To examine offences, penalties, and enforcement under the IT Act.5.
5. To analyze the role of electronic evidence and miscellaneous IT Act provisions

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the concepts of Internet, E-commerce, E-governance, and the role of electronic signatures in a free market economy, particularly in the Indian context.
CO2	Interpret and apply the legal aspects of electronic records, digital signatures, and certifying authority regulations in India, along with intellectual property rights protection in cyberspace.
CO3	Analyze international efforts and conventions, such as the Council of Europe Convention on Cyber Crimes, to understand global perspectives on cyberspace laws.
CO4	Evaluate the penalties, compensation mechanisms, offences, and judicial review processes under the Indian Information Technology Act.
CO5	Examine miscellaneous provisions of the IT Act, including electronic evidence handling, amendments, and rules for certifying authorities and website blocking.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	2	2	2	1	2	1	2	1	1	2
CO2	3	3	3	2	3	2	2	2	2	1	1	3
CO3	3	3	3	3	3	2	2	2	2	1	2	3
CO4	2	3	3	3	2	3	2	2	2	2	2	3
CO5	2	2	2	2	3	3	3	1	3	2	2	3

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA 1 and CA 2 are based on Assignment/Declared test/Quiz/Seminar/Group discussions/presentation, etc.
MSE is based on 50% of course content (first three units).
ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	Internet, E-Commerce and E-Governance with Reference to Free Market Economy Understanding Computers, Internet and Cyber Laws, Conceptual Framework of E-commerce: E-governance, The Role of Electronic Signatures in E-commerce with Reference to Free Market Economy in India.	**08**
2.	Law Relating to Electronic Records and Intellectual Property Rights in India Legal Aspects of Electronic Records/Digital Signatures, The Rules and Regulations of Certifying Authorities in India, Protection of Intellectual Property Rights in Cyberspace in India.	**08**
3.	International Efforts Relating to Cyberspace Laws and Cyber Crimes International Efforts Related to Cyberspace Laws, Council of Europe (COE) Convention on Cyber Crimes.	**08**
4.	Penalties, Compensation and Offences Under the Cyberspace and Internet in India Penalties, Compensation and Adjudication of Violations of Provisions of IT Act and Judicial Review Some Important Offences under the Cyberspace Law and the Internet in India, Other Offences under the Information Technology Act in India.	**08**
5.	Miscellaneous Provisions of It Act and Conclusions the Role of Electronic Evidence and the Miscellaneous Provisions of the IT Act, Information Technology Act as Amended up to2008, The Information Technology (Certifying Authorities) Rules, 2000, The Information Technology (Certifying Authorities) Rules, 2000, Ministerial Order on Blocking of Websites	**08**

Reference Books:

Sr. No.	Title	Author	Edition
1	Cyber Laws and It Protection	Harish Chander	PHI Publication
2.	Cyber Law and Information Security	Faiyaz Ahamad, KLSI	Dreamtech Press
3.	Information Technology Law: Law and Society	Murray	3rd Edition, Oxford University Press Oxford 2016
4.	Cyber Security	Sunit Belapure Nina Godbole	Wiley India Pvt. Ltd
5.	Cyber Laws and It Protection	Harish Chander	PHI Publication

Title of the Course: Open Elective-I C) Employability Skill Development Course Code: 25MCA1205	L	T	P	Credit
	3	--	-	3

Course Pre-Requisite:

Basic proficiency in English (reading, writing, speaking, and listening), Fundamental knowledge of computer science concepts, Familiarity with common office productivity tools (e.g., MS Office, email, presentations),Willingness to participate in group activities, discussions, and presentations

Course Description:

The *Employability Skill Development* course is designed to equip MCA students with the essential soft skills and professional competencies required to succeed in the modern workplace. This course emphasizes communication skills, teamwork, problem-solving, critical thinking, adaptability, and professional etiquette. It also covers resume writing, interview techniques, time management, and leadership skills to prepare students for recruitment processes and real-world challenges in the IT industry. Through interactive sessions, workshops, and practical exercises, students will develop a well-rounded personality and enhance their readiness for diverse career opportunities.

Course Objectives:

1. **Enhance communication skills** (verbal, non-verbal, and written) essential for professional and workplace interactions.
2. **Develop interpersonal and teamwork skills** required for collaborative work environments.
3. **Improve problem-solving and decision-making abilities** through real-world scenarios and activities.
4. **Build confidence and presentation skills** to effectively express ideas and opinions.
5. **Prepare students for job interviews and group discussions** through mock sessions and feedback.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Demonstrate effective oral and written communication skills suitable for academic and professional settings.
CO2	Exhibit teamwork, leadership, and interpersonal skills essential for collaborative work environments
CO3	Apply problem-solving and critical thinking techniques in workplace-like scenarios.
CO4	Prepare and present professional documents such as resumes, cover letters, and emails effectively.
CO5	Perform confidently in interviews, group discussions, and public speaking situations.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	3	2	2	2	-	2	1	2	1	2	2
CO2	3	3	3	2	2	2	2	2	3	3	2	2
CO3	2	2	3	2	3	1	2	2	3	1	2	3
CO4	1	2	2	1	1	1	2	3	3	2	3	2
CO5	1	2	2	1	2	-	2	2	3	2	3	2

Assessment Scheme:

Two components in semester Continuous Assessment (CA-1, CA-2), Mid Semester Examination (MSE) and End Semester Examination (ESE) having 20%,30% and 50% weightage respectively.

Assessment Component	Marks
CA1	10
MSE	30
CA2	10
ESE	50

CA 1 and CA 2 are based on Assignment/Declared test/Quiz/Seminar/Group discussions/presentation, etc.
MSE is based on 50% of course content (first three units).
ESE is based on 100% course content with 60-70% weightage for course content (last three units) covered after MSE.

Course Contents

Unit No.	Unit Title and Contents	Hours
1.	**Soft Skills & Communication basics:** Soft skills Vs hard skills, Skills to master, Interdisciplinary relevance, Global and national perspectives on soft skills, Resume, Curriculum vitae, How to develop an impressive resume, Different formats of resume Chronological, Functional, Hybrid, Job application or cover letter, Professional presentation- planning, preparing and delivering presentation, Technical writing.	**08**
2.	**Arithmetic and Mathematical Reasoning and Analytical Reasoning and Quantitative Ability:** Aspects of intelligence, Bloom taxonomy, multiple intelligence theory, Number sequence test, mental arithmetic (square and square root, LCM and HCF, speed calculation, reminder theorem). Matching, Selection, Arrangement, Verifications (Exercises on each of these types). Verbal aptitude (Synonym, Antonym, Analogy).	**08**
3.	**Grammar and Comprehension:** English sentences and phrases, Analysis of complex sentences, Transformation of sentences, Paragraph writing, Story writing, Reproduction of a story, Letter writing, précis writing, Paraphrasing and e-mail writing.	**08**
4.	**Skills for interviews:** Interviews- types of interviews, preparatory steps for job interviews, interview skill tips, Group discussion- importance of group discussion, types of group discussion, difference between group discussion, panel discussion and debate, personality traits evaluated in group discussions, tips for successful participation in group discussion, Listening skills- virtues of listening, fundamentals of good listening, Non-verbal communication-body movement, physical appearance, verbal sounds, closeness, time.	**08**

5.	**Problem Solving Techniques:** Problem solving model: 1. Define the problem, 2. Gather information, 3. Identify various solution, 4. Evaluate alternatives, 5. Take actions, 6. Evaluate the actions. Problem solving skills: 1. Communicate. 2. Brain storming, 3. Learn from mistakes	**08**

Text Books:

Sr. No.	Title	Author	Edition
1	Soft Skills- An integrated approach to maximize personality	R. Gajendra Singh Chauhan, Sangeeta Sharm	ISBN: 987-81-265-5639-7, First Edition 2016

Reference Books:

Sr. No.	Title	Author	Edition
1	English grammar and Composition	Wiley Wren and Martin,	S.Chand publications
2	A modern approach to verbal reasoning	R. S. Aggarwal	S. Chand publications
3	The Complete Book of Intelligence Test	Philip Carter	John Willey & Sons Ltd.-

Title of the Course: Object Design & Agile Development Lab Course Code: 25MCA1201L	L	T	P	Credit
	--	--	2	1

Course Pre-Requisite:

Basic programming knowledge. Understanding of software development lifecycle. Familiarity with object-oriented programming concepts

Course Description:

This course covers advanced software development methodologies, focusing on object-oriented programming and design using UML, distributed computing, and agile software development practices. It explores frameworks like Scrum and Kanban, design principles, and emphasizes software reuse, testing, and continuous integration for efficient and flexible software delivery.

Course Objectives:

1. Understand object-oriented concepts, design, and analysis using UML
2. Learn distributed object computing and middleware standards
3. Grasp agile methodologies, principles, and benefits
4. Apply Scrum and Kanban frameworks in project management
5. Develop skills in test-driven development, continuous integration, and risk-based testing.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Apply object-oriented concepts and modeling techniques using UML for software analysis and design.
CO2	Understand and evaluate distributed object computing and middleware standards such as COM, DCOM, and CORBA.
CO3	Explain the principles, values, and benefits of Agile software development methodologies.
CO4	Apply Scrum framework elements (roles, events, artifacts) to manage Agile software projects.
CO5	Apply Kanban framework and Agile testing practices including test-driven development and test automation.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	3	2	2	2	-	2	1	2	1	2	3
CO2	3	3	3	2	2	2	2	2	3	3	2	3
CO3	2	2	3	2	3	1	2	2	3	1	2	3
CO4	1	2	2	1	1	1	2	3	3	2	3	3
CO5	1	2	2	1	2	-	2	2	3	2	3	3

Assessment Scheme:

Two components of Continuous Assessment (CA-1, CA-2), End Semester Practical Oral Examination (POE), have 25%, 25% and 50% weightage, respectively.

Assessment Component	Marks
CA1	25
CA2	25
POE	--

CA1 and CA2 are based on 50% practical assignment.

Course Contents

Practical No.	List of Experiments	Hours
1	Designing a Real-World System Using UML Diagrams	2
2	Implementation of Core Design Patterns in Object-Oriented Programming	2
3	Developing a Distributed Client-Server Application Using CORBA	2
4	Integration of COM Components in a .NET Application	2
5	Modeling and Querying Data in an Object-Oriented Database	2
6	Refactoring Code Using SOLID Principles for Maintainability	2
7	Writing Effective User Stories and Acceptance Criteria in Agile	2
8	Simulating a Sprint Using Scrum Roles, Events, and Artifacts	2
9	Implementing a Kanban Board for Workflow Management	2
10	Test-Driven Development and Continuous Integration with Automated Testing	2

Text Books

Sr.No.	Title	Author	Edition
1	Object Oriented System Analysis	Sally Shlaer	Prentice Hall PTR
2	Object Oriented System Analysis and Design using UML	Simon Bennett	McGraw-Hill.
3	Agile Software Development with Scrum	Ken Schawber, Mike Beedle	Pearson
4	Agile Software Development, Principles, Patterns and Practices	Robert C. Martin	Prentice Hall

Reference Books:

Sr. No.	Title	Author	Edition
1	Agile Testing: A Practical Guide for Testers and Agile Teams	Lisa Crispin, Janet Gregory	Addison Wesley
2	Agile: The Complete Overview of Agile Principles and Practices	Paul VII	Agile Product Management
3	Agile Software Development, Principles, Patterns, and Practices	Robert Martin	Pearson New International Edition

Title of the Course: Data Analysis using Python Lab Course Code: 25MCA1206L	L	T	P	Credit
	--	--	2	1

Course Pre-Requisite:

Basic programming knowledge. Basic knowledge of programming logic and algorithm design (e.g., flowcharts or pseudocode), Understanding of computer fundamentals (data types, memory, input/output), exposure to any programming language (like C/C++/Java)

Course Description:

This course introduces students to the Python programming language for solving computational problems across a range of application domains. Beginning with foundational programming constructs like variables, control flow, and data structures, the course progresses to more advanced topics such as functions, object-oriented programming, and error handling. A strong emphasis is placed on writing clear, efficient, and maintainable code. Additionally, the course integrates the use of Python's powerful standard and third-party libraries for data analysis, scientific computing, and real-time applications

Course Objectives:

1. Understand the fundamental syntax and semantics of Python.
2. Apply conditional and iterative constructs to solve computational problems.
3. Implement classic searching and sorting algorithms using Python.
4. Work with Python's built-in data structures like lists, tuples, sets, and dictionaries.
5. Develop modular programs using functions, modules, and libraries.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Apply object-oriented concepts and modeling techniques using UML for software analysis and design.
CO2	Understand and evaluate distributed object computing and middleware standards such as COM, DCOM, and CORBA.
CO3	Explain the principles, values, and benefits of Agile software development methodologies.
CO4	Apply Scrum framework elements (roles, events, artifacts) to manage Agile software projects.
CO5	Apply Kanban framework and Agile testing practices including test-driven development and test automation.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	3	2	2	2		2	1	2	1	2	3
CO2	3	3	3	2	2	2	2	2	3	3	2	3
CO3	2	2	3	2	3	1	2	2	3	1	2	3
CO4	1	2	2	1	1	1	2	3	3	2	3	3
CO5	1	2	2	1	2		2	2	3	2	3	3

Assessment Scheme:

Two components of Continuous Assessment (CA-1, CA-2), End Semester Practical Oral Examination (POE), have 25%, 25% and 50% weightage, respectively.

Assessment Component	Marks
CA1	25
CA2	25
POE	50

CA1 and CA2 are based on 50% practical assignment.

POE is based on 100% Experiments

Course Contents

Practical No.	List of Experiments	Hours
1	Data Cleaning and Preprocessing Using Pandas	**2**
2	Exploratory Data Analysis (EDA) on a Real-World Dataset	**2**
3	Data Visualization Using Matplotlib and Seaborn	**2**
4	Performing Statistical Analysis Using SciPy	**2**
5	Correlation and Regression Analysis Using Python	**2**
6	Time Series Analysis and Forecasting Using Pandas and Statsmodels	**2**
7	Dimensionality Reduction Using PCA with Scikit-Learn	**2**
8	Clustering Data Using K-Means and Visualizing the Results	**2**
9	Classification Model Building Using Logistic Regression	**2**

10	Building and Evaluating a Machine Learning Pipeline for Data Analysis	**2**

Text Books			
Sr. No.	**Title**	**Author**	**Edition**
1	Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython	Wes McKinney	O'Reilly & Associates Inc.
2	Python Data Science Handbook	Jake VanderPlas	O'Reilly Media.

Reference Books:			
Sr. No.	**Title**	**Author**	**Edition**
1.	Python for Data & Analytics	Daniel Groner	--

Title of the Course: Artificial Intelligence Laboratory	**L**	**T**	**P**	**Credit**
Course Code: 25MCA1203L	**-**	**-**	**2**	**1**

Course Pre-Requisite:

Basic programming knowledge in any language (preferably Python or Java)., understanding of fundamental programming constructs like recursion, arrays, and control structures., Basic familiarity with algorithms and data structures.

Course Description:

This course offers practical hands-on experience with fundamental concepts and techniques of Artificial Intelligence through programming experiments. Students will study AI programming languages and environments such as Lisp, Prolog, Python, or Java. The course covers classical search algorithms, heuristic search, game theory methods like Min-Max and Alpha-Beta pruning, and constraint satisfaction problems. Through these experiments, students gain the skills to develop and analyze AI solutions for real-world problems such as puzzles, games, and optimization challenges.

Course Objectives:

1. Understand the basics of AI programming languages/environments including Lisp, Prolog, Python, or Java.
2. Explore existing AI applications and their functionalities.
3. Implement fundamental AI search algorithms like depth-first search, breadth-first search, and best-first search.
4. Develop AI solutions for classic problems such as Tic-Tac-Toe, 8-puzzle, traveling salesman, and 8-queens.
5. Apply advanced AI techniques like Min-Max search and Alpha-Beta pruning in game playing.

Course Outcomes:

CO	**After the completion of the course the student should be able to**
CO1	Understand and use AI programming languages/environments such as Lisp, Prolog, Python, or Java.
CO2	Analyze and explain existing AI applications and their problem-solving approaches.
CO3	Implement classical search algorithms including depth-first search and breadth-first search.
CO4	Solve heuristic search problems like 8 puzzle using best-first search techniques.
CO5	Develop game-playing AI using Min-Max search and Alpha-Beta pruning algorithms.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2			3		2		2		2	3
CO2	3	3		2					2	2		3
CO3	3	3	2	2	3		3	3	2	2	3	3
CO4	3	3	2	2	3		3	3	2	2	3	3
CO5	3	3	2	2	3		3	3	2	2	3	3

Assessment Scheme:

Two components of Continuous Assessment (CA-1, CA-2), End Semester Practical Oral Examination (POE), have 25%, 25% and 50% weightage, respectively.

Assessment Component	Marks
CA1	25
CA2	25
POE	50

CA1 and CA2 are based on 50% practical assignment.
POE is based on 100% Experiments

Course Contents

Practical No.	List of Experiments	Hours
1.	Write a Program to Implement Breadth First Search.	**02**
2.	Write a Program to Implement Depth First Search.	**02**
3.	Write a program to implement Hill Climbing Algorithm	**02**
4	Write a program to implement A* Algorithm	**02**
5.	Write a program to implement Tic-Tac-Toe game.	**02**
6	Write a Program to Implement 8-Puzzle problem using Python.	**02**
7	Write a Program to Implement Water-Jug problem using Python	**02**

8	Write a Program to Implement Travelling Salesman Problem using Python.	**02**
9	Write a Program to Implement Alpha-Beta Pruning using Python.	**02**
10	Write a Program to Implement 8-Queens Problem using Python.	**02**
11	Write a Program to Implement Tower of Hanoi using Python.	**02**

Text Books

Sr. No.	Title	Author	Edition
1	Artificial Intelligence: A Modern Approach	Dan W. Patterson	Pearson **4th Edition**, 2020

Reference Books:

Sr. No.	Title	Author	Edition
1.	Introduction to Artificial Intelligence and Expert Systems	Dan W. Patterson	Prentice Hall **1st Edition,** 1990
2.	Artificial Intelligence: Foundations of Computational Agents	David L. Poole and Alan K. Mackworth	Cambridge University Press **2nd Edition**, 2017

Title of the Course: Mini Project Course Code: 25MCA1207L	L	T	P	Credit
	-	--	2	1

Course Pre-Requisite:

1. Completion of core foundational courses in Computer Science and Engineering (such as Data Structures, Operating Systems, Computer Networks, etc.).
2. Basic knowledge of reading and interpreting academic papers or technical articles.
3. Familiarity with technical writing, implementation platforms and presentation tools (e.g., MS Word, LaTeX, PowerPoint).
4. A willingness to explore beyond the prescribed curriculum and independently investigate a selected topic.
5. Communication and comprehension skills sufficient to engage in discussions and deliver presentations in English.

Course Description:

The Mini Project course is designed to provide MCA students with practical experience in software development and project management. It enables students to apply theoretical knowledge gained in previous courses to solve real-world problems through the design and implementation of a software application or system. Students will work individually or in small teams under faculty supervision to complete the project lifecycle, including problem identification, requirements analysis, system design, implementation, testing, and documentation.

Course Objectives:

1. To develop and strengthen the students' practical skills in software design and development.
2. To promote analytical thinking, problem-solving, and innovation.
3. To provide experience in project planning, execution, and documentation.
4. To improve communication and teamwork abilities through group collaboration (if applicable).

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	Identify and define a real-world problem and propose a feasible software solution.
CO2	Apply appropriate tools, technologies, and methodologies for software development.
CO3	Develop a working prototype or application with proper testing and validation.
CO4	Prepare professional-quality technical documentation and project reports.
CO5	Present and demonstrate their project effectively to an evaluation panel.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	P011	P012
CO1	3	2	2	2	2		3	2	2	2	2	3
CO2	3	3	2	3	2		2		2	2		3
CO3	2	2	2		2	2	2		3	2	1	3
CO4		2	2		2		2		3	2	2	2
CO5	2	3	2	2	2	3	3		2	2	3	2

Assessment Scheme:

Two components of Continuous Assessment (CA-1, CA-2) will have 25% weightage, respectively and OE is having 50% weightage.

Assessment Component	Marks
CA1	25
CA2	25
OE	50