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		40
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PCC	25MCA1102	Data Structure and Algorithms using python	3	0	0	3	3	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
3	PCC	25MCA1103	Data Base Management System	3	0	0	3	3	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
4	BSC	25MCA1104	Probability and Statistics	3	1	0	4	4	CA1	10		40
									CA2	10		
									ESE	50	20	
5	PCC	25MCA1105	Java Programming	3	0	0	3	3	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
6	PCC	25MCA1102L	Data Structure & Algorithms using Python Laboratory	0	0	2	2	1	CA1	25		40
									CA2	25		
									POE	50	20	
7	PCC	25MCA1103L	Data Base Management System Laboratory	0	0	2	2	1	CA1	25		40
									CA2	25		
									POE	50	20	
8	PCC	25MCA1105L	Java Programming Laboratory	0	0	2	2	1	CA1	25		40
									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												

• 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		40
									MSE	30		
									CA2	10		
									ESE	50	20	
2	PCC	25MCA1202	Computer Networks	3	1	0	4	4	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
3	PCC	25MCA1203	Artificial Intelligence	3	0	0	3	3	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
4	PEC	25MCA1204	Program Elective I	3	0	0	3	3	CA1	10		40
									MSE	30		
									CA2	10		
									ESE	50	20	
5	OE	25MCA1205	Open Elective I	3	0	0	3	3	CA1	10		40
									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		40
									CA2	25		
									POE	50	20	
8	PCC	25MCA1203L	Artificial Intelligence Laboratory	0	0	2	2	1	CA1	25		40
									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 – 25 Total Credits – 20												

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	**8**

	Least Recently used(LRU). Sequences and convergence, Infinite 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	**8**

	Python: traversing and searching in lists, stack push/pop using list, 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**

3	**Correlation** 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.