MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)
#14, Ramohalli Cross, Kumbalagodu, Mysore Road, Bengaluru–560074



Master of Computer Applications

I Semester Scheme and Syllabus

(2026 Scheme)

VISION

To empower young minds through technology, research and innovation, to produce technically competent and socially responsible professionals in higher education.

MISSION

1. To deliver excellence in education through innovative teaching, impactful research, and continuous skill development, preparing students to meet global challenges with technical expertise and ethical responsibility.
2. To foster a transformative learning environment that integrates technology, research and practical experience, empowering students to become skilled professionals and socially conscious leaders.
3. To cultivate a culture of lifelong learning and professional excellence by encouraging creativity, research, and community engagement, equipping students with the skills to thrive in a dynamic world.
4. To provide a holistic educational experience that combines advanced technology, hands-on research, and community-focused learning, shaping students into competent, ethical professionals who contribute positively to society.

QUALITY POLICY

Rajarajeswari College of Engineering is committed to imparting quality technical education that nurtures competent, ethical professionals with global relevance. We ensure academic excellence through a dynamic, outcome-based curriculum, experienced faculty, and cutting-edge infrastructure. Continuous improvement is driven by innovation, research and strong industry collaboration. We foster holistic development and a progressive environment that supports lifelong learning, teamwork, and professional growth.

CORE VALUES

Academic Excellence, Integrity, Innovation, Global Competence, Continuous Improvement.

Department of Master of Computer Applications

Vision

To be a department of excellence in technical education, widely known for the development of application developers, IT professionals, entrepreneurs, researches creates benefit all of humanity.

Mission

To ensure diverse knowledge integration with industry for creative learning and application.
To nuture talent in usage of application of Digital Intelligence and use of Power of technology.
To create entrepreneurs thro incubating technology businesses and committee to create and sustaining to be societal relevant education
To ensure the student connect at personal level and focus on to holistic development with Intellectual, Emotional, Spiritual and Behavioral traits.

Programme Educational Objectives of MCA programme are:

BCA programme of RRCE will prepare its students:

PEO 1: To progress into the software industry, academia, research, entrepreneurial pursuit, government, consulting firms, and other recent trends in technology enabled services.

PEO 2: To achieve peer recognitions an individual or team; by adopting ethics and professionalism and communicating effectively to perform well in cross-culture and inter-disciplinary teams.

PEO 3: To continue lifelong professional development in computing for the benefit of self- and societal growth

To progress into the software industry, academia, research, entrepreneurial pursuit, government, consulting firms, and other recent trends in technology enabled services.

To achieve peer recognitions an individual or team; by adopting ethics and professionalism and communicating effectively to perform well in cross-culture and inter-disciplinary teams.

Program Outcomes

PO 1 Knowledge: Able to understand and apply the fundamental principles, concepts and methods in diverse areas of computer applications, mathematics, statistics etc.

PO 2 Problem analysis and Development of Solutions: Identify, formulate, research literature and analyze real- time problems. Attain substantiated conclusions to solve the problems using fundamental principles of mathematics, computing sciences by adopting various tools and techniques.

PO 3 Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO 4 Ethics and Social Responsibility: Understand and commit to professional ethics and cyber regulations, responsibilities and norms of professional computing practice.

PO 5 Communication and Team Building: Possess interpersonal skills and communicate effectively with the professionals and with society at large on system activities. Able to comprehend and write effective reports, design documentation, make effective presentations, and give/understand instructions.

PO 6 Project Management and Finance: Innovation, Employability and Entrepreneurial skills Identify opportunity; pursue that opportunity to create value and wealth for the betterment of the individual and society at large. Develop the capacity to study and research independently that will help to develop skills for transition to employment in hardware/software companies.

PO 7 Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO 8 Life Long Learning: Recognize the need and have the ability, to engage in continuous reflective learning in the context of technological advancement. Create, select, adapt and apply appropriate techniques, resources, and computing tools to complex computing activities. Able to learn, adapt and apply emerging tools and technologies to meet the demand.

INDEX

I Semester			
Sl. No.	**Course Code**	**Course Title**	**Page No.**
1.	P26MCA101	Mathematical Methods	01
2.	P26MCA102	Database Management Systems	03
3.	P26MCA103	AI and Applications	05
4.	P26MCA104	Data Strtuctures and Alogorithms	07
5.	P26MCA105	Python Programming and Applications	09
6.	P26MCA106	Java Programming	12
7.	P26MCAL107	Data Structures and Algorithms Laboratory	14

MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Master of Computer Applications

Scheme of Teaching and Examinations – 2026 Scheme

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

I Semester

Academic Year: 2026-27

S. No	Course Category and Course Code		Course Title	TD / PSB	Teaching Hours / Semester					Examination				Credits
					Lecture	Tutorial	Practical	TW + SL	Total Hours / Sem	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S						
1	BSC	P26MCA101	Mathematical Methods	MCA	42	0	0	48	90	3	50	50	100	3
2	PCC	P26MCA102	Database Management Systems	MCA	42	0	0	48	90	3	50	50	100	3
3	PCC	P26MCA103	AI and Applications	MCA	42	0	0	48	90	3	50	50	100	3
4	PCC	P26MCA104	Data Structures and Algorithms	MCA	42	0	0	48	90	3	50	50	100	3
5	IPCC	P26MCA105	Python Programming and Applications	MCA	42	0	28	50	120	3	50	50	100	4
6	PCC	P26MCA106	JAVA Programming	MCA	42	0	0	48	90	3	50	50	100	3
7	PCCL	P26MCAL107	Data Structures and Algorithms Laboratory	MCA	0	0	28	02	30	3	50	50	100	1
TOTAL									**600**	**-**	**350**	**350**	**700**	**20**
8	NCMC	P26MCA108	Research Methodology and IPR (Online)			**ONLINE COURSES (Online.vtu.ac.in)**								**PP**
9	SKILL	P26SKL100	Generative AI & Prompt Engineering	MCA							100	-	100	1

HoD **Dean-Academics** **Principal**

Semester - I					
Mathematical Methods **Category: BSC**					
Course Code	:	**P26MCA101**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:0:48	SEE	:	50 Marks
Total Hours / Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To introduce basic mathematical concepts such as sets, logic, relations, and functions used in computer applications.
2	To develop problem-solving skills using basic counting techniques and graph concepts
3	To provide fundamental knowledge of vectors and matrices for solving computational problems
4	To introduce basic concepts of probability and statistics for data analysis and computing applications
5	To develop an understanding of basic calculus and optimization concepts relevant to computer applications and machine learning.

Module 1	**No. of Hrs**
Sets: Basic concepts of sets, set representation, set operations and Venn diagrams. **Logic:** Propositions, logical connectives and truth tables. **Relations:** Basic concepts and properties of relations. **Functions:** Definition, types of functions and basic applications.	09
Module 2	**No. of Hrs**
Counting: Basic counting principles, permutations and combinations. **Graphs:** Introduction to graphs, terminology and types of graphs. **Graph Representation:** Basic graph representation and applications. **Paths and Trees:** Paths, cycles and basic concepts of trees.	09
Module 3	**No. of Hrs**
Vectors: Basic concepts of vectors and vector operations. **Matrices:** Definition, types of matrices and basic matrix operations. **Determinants:** Evaluation and properties of determinants. **Inverse:** Inverse of a matrix. **Linear Equations:** Solving systems of linear equations using matrices.	08
Module 4	**No. of Hrs**
Probability: Basic concepts of probability, sample space and events. **Conditional Probability:** Conditional probability and related problems. **Bayes' Theorem:** Basic applications of Bayes' theorem. **Statistics:** Mean, median and mode. **Measures of Dispersion:** Variance and standard deviation. **Correlation:** Basic concept and interpretation of correlation.	08
Module 5	**No. of Hrs**
Functions and Limits: Basic concepts of functions and limits. Differentiation: Basic differentiation and its applications. **Partial Derivatives:** Basic concepts and simple applications. **Maxima and Minima:** Finding maxima and minima of simple functions. Optimization: Introduction to gradient and basic optimization concepts.	08

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply sets, logic, relations and functions to solve basic mathematical problems.
CO2	Apply counting techniques and graph concepts to solve computational problems.
CO3	Solve basic problems involving vectors, matrices and linear equations.
CO4	Apply probability and statistical measures to analyze basic data.
CO5	Apply basic calculus and optimization concepts to solve computing problems.

Text Books	
1	Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw-Hill Education.
2	Seymour Lipschutz and Marc Lipson, Discrete Mathematics, Schaum's Outline Series, McGraw-Hill Education.

Reference Text Books	
1	Richard Johnsonbaugh, *Discrete Mathematics*, Pearson.
2	Howard Anton and Chris Rorres, *Elementary Linear Algebra*, Wiley.
3	S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*, Sultan Chand & Sons

Web links and Video lectures (e-Resources)

1. **MIT OpenCourseWare – Mathematics for Computer Science**
   https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-spring-2015/
2. **Khan Academy – Statistics and Probability**
   https://www.khanacademy.org/math/statistics-probability
3. **MIT Open Learning Library – Mathematics for Computer Science**
   https://openlearninglibrary.mit.edu/courses/course-v1:OCW+6.042J+2T2019/about
4. **NPTEL – Discrete Mathematics:** NPTEL – Discrete Mathematics
5. **Wolfram MathWorld – Mathematics Reference:** Wolfram MathWorld

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	50
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any Two Assessments as per regulations	20	10	
SEE	Semester End Examination	100	50	50
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping:

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

Semester - I					
Database Management Systems **Category: PCC**					
Course Code	:	**P26MCA102**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To introduce the fundamental concepts of database systems, DBMS architecture, and Entity-Relationship modeling.
2	To develop skills in SQL, database design, normalization, and transaction processing
3	To introduce advanced SQL, PL/SQL, and NoSQL database concepts for modern database applications
4	To develop practical knowledge of MongoDB operations, querying, aggregation, indexing, replication, and sharding
5	To provide an understanding of database security, SQL injection, backup, recovery, and database failure management.

Module – 1	**No. of Hrs**
Introduction to DBMS: Characteristics of database approach, Advantages and applications of DBMS, Three-schema architecture and data independence Entity Relationship Model: Entity types and entity set, Attributes and keys, Relationship types and constraints, Weak entities, ER diagrams for real-world applications, ER-torelational mapping.	09
Module – 2	**No. of Hrs**
SQL Fundamentals: SQL data types and constraints, DDL, DML, DCL, TCL commands, Simple and nested querie-s, Joins and set operations, Aggregate functions and grouping, Views. Database Design: Functional dependencies, Normalization (1NF, 2NF, 3NF, BCNF) Transactions Processing Concepts: Transaction concepts, ACID properties.	08
Module – 3	**No. of Hrs**
Advanced SQL and PL/SQL: Stored procedures, functions and triggers, Cursors and exception handling NoSQL: Introduction to NoSQL and need for NoSQL databases, Characteristics of NoSQL systems, Types of NoSQL databases – Key-value stores, Document databases, Columnfamily databases and Graph databases, Storage architecture basics.	08
Module – 4	**No. of Hrs**
MongoDB: Introduction to MongoDB, CRUD Operations, Querying, Aggregation, Indexing, Replication, Sharding.	08
Module – 5	**No. of Hrs**
Database Security and Recovery: Introduction to Database Security, Discretionary Access Control, Mandatory Access Control and Role-Based Access Control, SQL Injection, Database Backup and Recovery Concepts, Types of Database Failures, Recovery Techniques based on Deferred and Immediate Update, Shadow Paging.	09

Course Outcomes: At the end of this course, students are able to:	
CO1	Apply the basic concepts of Database Management.
CO2	Write SQL, PL/SQL queries for a given problem scenario.
CO3	Apply NoSQL data modeling concepts.
CO4	Perform CRUD operations using MongoDB.
CO5	Build a database for any given use-case.

Text Books	
1	Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition, 2016
2	ShashankTiwari, Professional NoSQL, Wiley India/Wrox Publication, 1st Edition, 2011.
3	Kristina Chodorow, MongoDB: The Definitive Guide, O'Reilly Media, 1st Edition

Reference Text Books	
1.	Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, McGraw Hill, 7th Edition.
2.	Pramod J. Sadalage and Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional.

Web links and Video lectures (e-Resources)
MySQL Documentation: https://dev.mysql.com https://w3schools.com/sql **MongoDB Documentation:** https://www.mongodb.com/resources/products/fundamentals/mongodb-tutorials **Interactive SQL:** https://sqlbolt.com/

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	50
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any Two Assessments as per regulations	20	10	
SEE	Semester End Examination	100	50	50
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping:

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

Semester - I					
AI and Applications **Category: PCC**					
Course Code	:	**P26MCA103**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objectives is	
1	To introduce the fundamental concepts of Artificial Intelligence, intelligent agents, and problem-solving environments
2	To develop an understanding of search strategies, optimization techniques, and knowledge representation methods used in AI.
3	To introduce adversarial search, logical reasoning, and basic machine learning concepts for intelligent decision-making.
4	To develop awareness of explainable, fair, transparent, and responsible Artificial Intelligence.
5	To provide knowledge of AI applications in real-world domains such as healthcare, finance, agriculture, robotics, and smart cities.

Module – 1	**No. of Hrs**
Introduction and Intelligent Agents: Introduction; What is Artificial Intelligence? The Foundations of Artificial Intelligence; The History of Artificial Intelligence; The State of the Art; Intelligent Agents; Agents and Environments; The Nature of Environments; The Structure of Agents.	08
Module – 2	**No. of Hrs**
Solving Problems by Searching: Problem-Solving Agents; Example Problems; Searching for Solutions; Uninformed Search Strategies; Informed (Heuristic) Search Strategies; Local Search Algorithms and Optimization Problems; Search in Complex Environments; Online Search Agents and Unknown Environments.	08
Module – 3	**No. of Hrs**
Adversarial Search and Knowledge-Based Systems: Adversarial Search and Games; Optimal Decisions in Games; Alpha–Beta Pruning; Knowledge-Based Agents; Propositional Logic; First-Order Logic; Inference in First-Order Logic.	08
Module – 4	**No. of Hrs**
Machine Learning Concepts and Responsible AI: Introduction to Machine Learning, Relationship between Artificial Intelligence, Machine Learning and Deep Learning, Types of Machine Learning (Supervised, Unsupervised and Reinforcement Learning), AI Model Development Lifecycle, Model Training, Validation and Testing, Challenges in Machine Learning, Explainable AI (XAI), Responsible AI, Bias, Fairness, Transparency, Ethics and Risks of Artificial Intelligence.	09
Module – 5	**No. of Hrs**
Artificial Intelligence Applications: Introduction to Natural Language Processing and Large Language Models, Computer Vision, Robotics and Intelligent Autonomous Systems, Multi-Agent Systems, AI Applications in Healthcare, Finance, Agriculture, Smart Cities and Transportation, Responsible and Ethical AI.	09

Course Outcomes: At the end of this course, students are able to:	
CO1	Explain the fundamental concepts of artificial intelligence, intelligent agents, agent architectures, and the characteristics of problem-solving environments.
CO2	Apply search strategies, optimization techniques, and knowledge representation methods to solve artificial intelligence problems.
CO3	Analyse adversarial search, logical inference, and machine learning approaches for intelligent decision-making in diverse application domains.
CO4	Apply explainability, bias mitigation, fairness, ethics, responsible AI principles, and performance measures in the development and assessment of artificial intelligence solutions.
CO5	Analyse and explain the use of Artificial Intelligence techniques in solving real-world problems.

Text Books	
1	Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education, 4th Global Edition, 2021.
2	Tom Taulli, Artificial Intelligence Basics: A Non-Technical Introduction, 2nd Edition, A press, 2023.

Reference Text Books	
1	Artificial Intelligence: Foundations of Computational Agents, David L. Poole and Alan K. Mack worth, Cambridge University Press, 2017.
2	Deep Learning, Ian Goodfellow, YoshuaBengio, and Aaron Courville, MIT Press, 2016.
3	Machine Learning, Tom M. Mitchell, McGraw-Hill Education, 1997.
4	Artificial Intelligence, McGraw-Hill Education, 3rd Edition.

Web links and Video lectures (e-Resources)
NPTEL course Artificial Intelligence: Concepts and Techniques https://onlinecourses.nptel.ac.in/noc25_cs159/preview • Course AI for Everyone (Deep earning.ai) AI for Everyone • Geeks for Geeks Introduction to AI and Its Application https://www.geeksforgeeks.org/artificial-intelligence/artificial-intelligence/

ASSESSMENT STRUCTURE:
The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	50
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any Two Assessments as per regulations	20	10	
SEE	Semester End Examination	100	50	50
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping:

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

Semester - I					
Data Structures and Alogorithms **Category: PCC**					
Course Code	:	**P26MCA104**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To understand the fundamental concepts of data structures, including arrays, linked lists, stacks, queues, trees, graphs, and hashing techniques.
2	To analyze algorithms using mathematical analysis and asymptotic notations to determine their time and space complexities.
3	To apply appropriate searching, sorting, and traversal techniques to solve computational problems efficiently.
4	To develop solutions using algorithmic paradigms such as brute force, decrease-and-conquer, divide-and-conquer, greedy methods, and transform-and-conquer.
5	To implement and evaluate advanced data structures and algorithms, including heaps, AVL trees, 2-3 trees, minimum spanning trees, and shortest-path algorithms.

Module – 1	**No. of Hrs**
Foundations of Data Structures & Algorithmic Analysis: Data Structures Introduction: Fundamental Data Structures. Linear Array: Arrays, Dynamically Allocated Array, Linked Memory: Singly linked lists and chains, Doubly linked lists. Algorithms Analysis Introduction: What is an Algorithm? Fundamentals of Algorithmic Problem Solving, The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes.	08
Module – 2	**No. of Hrs**
Stacks, Queues, and Mathematical Analysis of Algorithms: Stacks and Queues: Stacks, Stacks Using Dynamic Arrays, Queues, Circular queues Using Dynamic Arrays, Evaluation of expression. Mathematical Analysis: Mathematical Analysis of Non-recursive Algorithms, Mathematical Analysis of Recursive Algorithms	08
Module – 3	**No. of Hrs**
Tree Structures, Searching & Hashing: Trees: Introduction, Binary Trees, Binary Tree Traversal, Binary Search Trees (BST). Searching Algorithms: Sequential Search and Brute-Force String Matching, and Decrease and Conquer - Binary Search. Hashing: Introduction and Static Hashing.	08
Module – 4	**No. of Hrs**
Graph Algorithms & Sorting Paradigms: Graphs: The Graph Abstract Data Type. Graph Traversals: Depth-First Search (DFS) and Breadth-First Search (BFS). Brute-Force Sorting: Selection Sort and Bubble Sort. Divide-and-Conquer Sorting: Merge Sort and Quick Sort	09
Module – 5	**No. of Hrs**
Greedy, Transform-and-Conquer, and Balanced Trees: The Greedy Method: Minimum Spanning Trees via Prim's Algorithm. Single-Source Shortest Path via Dijkstra's Algorithm. Transform-and-Conquer Structures: Heaps: Heaps and Heap Sort, Balanced Search Trees: AVL Trees, 2-3 Trees.	09

Course Outcomes: At the end of this course, students are able to:	
CO1	Explain fundamental data structures and their operations.
CO2	Analyze algorithms using time and space complexity.
CO3	Apply searching, sorting, hashing, and tree techniques.
CO4	Implement graph algorithms and sorting paradigms.
CO5	Apply greedy and transform-and-conquer algorithms.

Text Books	
1	Ellis Horowitz and SartajSahni, Fundamentals of Data Structures in C, 2nd Edition (2023), Universities Press(India) Private limited.
2	AnanyLevitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition (2024), Pearson Education, Limited.

Reference Text Books	
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 4th Edition (2022), MIT Press.
2	Reema Thareja, Data Structures using C, 3rd Edition (2023), Oxford University Press.

Web links and Video lectures (e-Resources)
W1: National Programme on Technology Enhanced Learning (NPTEL), Programming, Data Structures and Algorithms in C, Indian Institute of Technology Madras. https://nptel.ac.in/courses/106106127
W2: National Programme on Technology Enhanced Learning (NPTEL), Design and Analysis of Algorithms, Indian Institute of Technology Madras. https://nptel.ac.in/courses/106106131
W3: Massachusetts Institute of Technology (MIT) Open Course Ware, Introduction to Algorithms (6.006), Department of Electrical Engineering and Computer Science. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/
W4: Visualgo Data Structure Visualizations, Interactive animations of sorting, search trees, and graphs. https://visualgo.net/en

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	50
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any Two Assessments as per regulations	20	10	
SEE	Semester End Examination	100	50	50
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping:

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

Semester - I					
Python Programming and Applications **Category: IPCC**					
Course Code	:	**P26MCA105**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:28:50	SEE	:	50 Marks
Total Hours / Semester	:	120	Total	:	100 Marks
Credits		4	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To understand the fundamentals and syntax of Python programming.
2	To develop problem-solving skills using conditional statements, loops and functions.
3	To work effectively with Python's built-in data structures.
4	To develop modular, file-based and object-oriented Python programs.
5	To apply Python programming to simple real-world applications using standard libraries.

Module – 1	**No. of Hrs**
PYTHON FUNDAMENTALS : Introduction to Python – Features and Applications of Python – Python Interpreter – Interactive and Script Modes – Basic Syntax – Keywords and Identifiers – Variables – Comments – Indentation – Data Types – Numbers – Boolean – Strings – Type Conversion – Operators – Arithmetic – Relational – Logical – Assignment – Bitwise – Membership – Identity – Operator Precedence – Input and Output – input() – print() – String Formatting.	09
Module – 2	**No. of Hrs**
CONTROL STRUCTURES AND FUNCTIONS : Conditional Statements – if – if-else – if-elif-else – Nested if – Iterative Statements – for – while – range() – Nested Loops – Loop Control Statements – break – continue – pass – Functions – Function Definition and Calling – Parameters and Arguments – Positional and Keyword Arguments – Default Arguments – Variable-Length Arguments – Return Values – Scope of Variables – Recursive Functions – Lambda Functions.	09
Module – 3	**No. of Hrs**
STRINGS AND PYTHON DATA STRUCTURES : Strings – Indexing – Slicing – String Operators – String Methods – Lists – List Operations – List Methods – Nested Lists – Tuples – Tuple Operations and Methods – Sets – Set Operations – Union – Intersection – Difference – Dictionaries – Keys and Values – Dictionary Methods – Nested Dictionaries – List Comprehension – Set Comprehension – Dictionary Comprehension.	08
Module – 4	**No. of Hrs**
FILE HANDLING, EXCEPTIONS, MODULES AND OOP : File Handling – File Types – Opening and Closing Files – File Modes – Reading and Writing Files – Appending Data – with Statement – CSV Files – Errors and Exceptions – try – except – else – finally – Raising Exceptions – Modules – Importing Modules – Built-in Modules – User-defined Modules – Packages – Object-Oriented Programming – Classes and Objects – Constructor – Attributes and Methods – self – Inheritance – Polymorphism – Encapsulation.	08
Module – 5	**No. of Hrs**
PYTHON APPLICATIONS : Python Standard Libraries – math – random – date-time – os – Regular Expressions – Pattern Matching – match() – search() – findall() – Data Processing using Python – CSV Data Processing – Basic Statistics – Introduction to SQLite – Database Connection – Creating Tables – Insert – Select – Update – Delete – Introduction to Data Visualization – Matplotlib – Line Chart – Bar Chart – Pie Chart – Basic Python Application Development.	08

Sl. No	**List of experiments**
1.	Write a Python program to perform basic arithmetic operations on two numbers.
2.	Write a Python program to calculate Simple Interest, Compound Interest, and Total Amount.
3.	Write a Python program to find the largest of three numbers using conditional statements.
4.	Write a Python program to find factorial, Fibonacci series, and prime numbers using loops.

5.	Write a Python program to create a menu-driven calculator using functions.
6.	Write a Python program to perform string operations such as reverse, palindrome, vowel count, and character count.
7.	Write a Python program to perform list, tuple, and set operations.
8.	Write a Python program to create a student record using dictionaries and search/display student details.
9.	Write a Python program for file handling to create, read, write, append, and search data in a text file.
10.	Write a Python program to demonstrate exception handling using try, except, else, and finally.
11.	Write a Python program to create and use a user-defined module for mathematical operations.
12.	Write a Python program to demonstrate Object-Oriented Programming using classes, objects, constructors, and inheritance.
13.	Write a Python program using regular expressions to validate email addresses and mobile numbers.
14.	Write a Python program to perform SQLite database operations – create, insert, display, update, and delete records.
15.	Develop a mini Python application such as Student Management System, Library Management System, or Expense Tracker.

Course Outcomes: At the end of the course, the students will be able to	
CO1	Explain Python programming fundamentals, syntax, data types and operators.
CO2	Develop Python programs using decision-making, loops and functions.
CO3	Apply Python strings and built-in data structures to solve computational problems.
CO4	Develop modular, file-based and object-oriented Python applications with appropriate exception handling.
CO5	Apply Python libraries, regular expressions and database connectivity to develop simple real-world applications.

Text Books	
1.	**Eric Matthes**, Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 3rd Edition, No Starch Press, 2023.
2.	**Mark Lutz**, Learning Python, 6th Edition, O'Reilly Media, 2025.

Reference Text Books	
1.	**Allen B. Downey**, Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly Media, 2015.
2	**Mark Lutz**, Programming Python: Powerful Object-Oriented Programming, O'Reilly Media.
3	**Wes McKinney**, Python for Data Analysis, O'Reilly Media.

Web links and Video lectures (e-Resources)
1. Python Official Documentation – https://docs.python.org/3/ 2. Python Official Website – https://www.python.org/ 3. Python Tutorial – https://docs.python.org/3/tutorial/ 4. Python Standard Library – https://docs.python.org/3/library/

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	**50/2 = 25**
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any One Assessment	10	10	
Laboratory	Lab Conduction & Record	Evaluating each expt. for 10 mark.	10	**25**
	Lab Internal Test	50	15	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping:

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

Level 3 - High, Level 2 - Moderate, Level 1 - Low

Semester - I					
Java Programming **Category: PCC**					
Course Code	:	**P26MCA106**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:0:0:48	SEE	:	50 Marks
Total Hours / Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objectives is	
1	To understand Java programming fundamentals, classes, objects, methods, constructors, and control statements.
2	To apply inheritance, interfaces, method overriding, and packages to develop modular Java applications.
3	To implement exception handling and multithreading concepts for robust and concurrent Java applications.
4	To develop web applications using Servlets and JSP with request handling, sessions, and cookies.
5	To understand Spring Framework concepts, including IoC, Dependency Injection, and application architecture.

Module – 1	**No. of Hrs**
Java Programming Fundamentals: An overview of Java, Data Types and Operators, Program Control Statements, Introducing Classes, Objects and Methods. Methods, Classes, Constructors, Access control, Pass Objects to Methods, How Arguments are passed, Returning Objects, Method Overloading, Overloading Constructors.	08
Module – 2	**No. of Hrs**
Inheritance, Interfaces and Packages: Inheritance, using super to Call Superclass constructors, using super to Access Superclass Members, creating a Multilevel Hierarchy, Superclass References and Subclass Objects, Method Overriding, Using Abstract Classes. Interface Fundamentals, Creating an Interface, implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be extended, Nested Interfaces, Package Fundamentals, Packages and Member Access, Importing Packages.	08
Module – 3	**No. of Hrs**
Exception Handling and Multi-Threading Programming: The Exception Hierarchy, Exception Handling Fundamentals, the Consequences of an Uncaught Exception, using Multiple catch clauses, Catching subclass Exceptions, Throwing an Exception, Throwable, using finally, using throws; Multithreading: The Thread Class and Runnable Interface, Creating Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), Suspending, Resuming and stopping Threads.	08
Module – 4	**No. of Hrs**
Servlet Programming: Servlet Structure, Packaging, Lifecycle, HTTP Request and response, Handling client request, Form data, HTTP status request headers, HTTP Status codes, HTTP response headers, Handling cookies, Session tracking. JSP Programming: Need of JSP, Basic syntax, Scripting elements, Limiting Java code in JSP, JSP expression, JSP directives, JSP attributes.	09
Module – 5	**No. of Hrs**
Spring Framework: Introducing the Spring Framework, Problems with the Traditional Approach to J2EE, Lightweight Frameworks, Spring's Values, Spring in Context, Inversion of Control and Dependency Injection, Architecting Applications with Spring.	09

Course Outcomes: At the end of this course, students are able to:	
CO1	Demonstrate object-oriented programming concepts.
CO2	Apply Java constructs to obtain solutions.
CO3	Develop reusable and efficient Java code.
CO4	Build a Java based client-server application.
CO5	Implement Java application using Spring framework.

Text Books	
1.	Kathy Sierra and Bert Bates, Head First Design Patterns, O'Reilly Media.
2.	Joseph B. Ottinger Jeff Linwood Dave Minte, Beginning Hibernate 6 Java Persistence from Beginner to Pro, 5th Edition, Apress.
3.	Jason Brittain and Brian D. Eubanks, Java Servlet and JSP Programming, O'Reilly Media, 1st Edition, 2001

Reference Text Books	
1.	Kathy Sierra and Bert Bates, Head First Design Patterns, O'Reilly Media.
2.	Joseph B. Ottinger Jeff Linwood Dave Minte, Beginning Hibernate 6 Java Persistence from Beginner to Pro, 5th Edition, Apress.

Web links and Video lectures (e-Resources)
1. Java Programming Fundamentals: www.java.com
2. Java Coding: https://www.programiz.com/java-programming
3. JSP Fundamentals: https://jakarta.ee/specifications/pages/
4. Spring Framework: https://spring.io/projects/spring-framework

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 50% of 50 marks, i.e., 25 marks. To pass the SEE, a student must score at least 40% of 50 marks, i.e., 20 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 50 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	40 (Average of Best two Assessments)	50
	Internal Assessment-2	50		
	Internal Assessment-3	50		
AAT	Any Two Assessments as per regulations	20	10	
SEE	Semester End Examination	100	50	50
Grand Total				**100**

SEMESTER END EXAMINATION (SEE):

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy level

CO-PO Mapping:

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

Semester - I					
Data Structures and Algorithms Laboratory **Category: PCCL**					
Course Code	:	**P26MCAL107**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	0:0:28:02	SEE	:	50 Marks
Total Hours / Semester	:	30	Total	:	100 Marks
Credits		1	SEE Duration	:	3 Hrs

Course Learning Objectives is	
1.	To make use of appropriate data structures and algorithmic techniques to solve complex, real- world computational problems.
2.	Develop code to perform required operations and validate results.
3.	To make use of appropriate programming tools and development environments for implementing, testing, and debugging data structure and algorithm applications.
4.	To build a technical document consisting of experiment setup, design, implementation, and results with inferences.

1. The laboratory syllabus consists of PART-A and PART-B. While PART-A has 6-8 conventional experiments, PART-B has 6 typical open-ended experiments. The maximum marks for the laboratory course are 100.
2. Both PART-A and PART-B are considered for CIE and SEE.
3. Students have answer 1(one) question from PART-A and 1(one) question from PART-B.
a. The questions set for SEE shall be from among the experiments under PART-A. It is evaluated for **80marks** out of the maximum 100 marks.
b. The open-ended question set for SEE shall be any other open-ended question and It shall be evaluated for **20marks.**
For Continuous Internal Evaluation (CIE) during the semester, class work shall typically include open-ended questions from **Part-B**, along with other similar questions designed to enhance the students' skills.

PART A	
S.No	**List of Experiments**
1.	Implement a Program in C for converting an Infix Expression to Postfix Expression.
2.	Design, develop, and execute a program in C to simulate the working of a Queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display.
3.	Develop a C program to simulate the working of a singly linked list providing the following operations: a. Display & Insert b. Delete from the beginning/end c. Delete a given element.
4.	Develop a C program to Implement the following searching techniques: a. Linear Search b. Binary Search.
5.	Develop a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
6.	Develop a C Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
PART-B : OPEN ENDED EXPERIMENTS Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course coordinator. It encourages creativity, critical thinking, and inquiry-based learning.	
1	**Concept 1: Stack Applications** Design and develop an application using stack operations for solving a suitable computational problem.
2	**Concept 2: Queue Applications** Develop an application using queue-based data structures to perform efficient data management operations.
3	**Concept 3: Doubly Linked List Applications** Design and implement an application using doubly linked list concepts for dynamic data storage and manipulation.
4	**Concept 4: Sorting Techniques and Performance Analysis** Implement suitable sorting techniques for a given problem and analyze the performance based on input size and execution time generator.

5	**Concept 5: Graph Algorithms Applications** Apply graph-based algorithms to solve problems related to shortest path.
6	**Concept 6: Recursive / Non recursive Techniques** Develop an application using recursive or non-recursive techniques to solve computational problems.

Course Outcomes: At the end of the course, the students will be able to	
CO1	Implement fundamental linear data structures such as stacks, queues and linked lists.
CO2	Apply searching and sorting techniques to solve computational problems.
CO3	Implement graph algorithms for solving connectivity and shortest-path problems.
CO4	Develop recursive and non-recursive solutions for computational problems.

ASSESSMENT STRUCTURE:
The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	**Lab Conduction & Record**	**Evaluating Each Expt. for 10 marks * 12 Expts.**	**15**	**50**
	Internal Test – 1	**50**	**15**	
	Internal Test – 2	**50**	**20**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

CO-PO Mapping

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