

OMEGA INSTITUTE OF SCIENCE AND TECHNOLOGY

MASTER OF COMPUTER APPLICATIONS

(MCA)

CHOICE BASE CREDIT SYSTEM(CBCS) SYLLABI

SEMESTER I AND II

With effect from Academic Year 2026-2027(Admitted)

**Under R26 Academic Regulations
(Applicable to Students Admitted from the
Academic Year 2026-2027)**



FACULTY OF INFORMATICS

OMIST(A)

Effective from A.Y: 2026-2027

(Admitted Batch)

MCA Program Outcomes

PO 1: Computational Knowledge

Apply knowledge of computing fundamentals, mathematics, and domain-specific expertise to model, design, and develop solutions for well-defined problems and requirements.

PO 2: Problem Analysis

Solve complex computing problems by exploring relevant literature and providing validated solutions, using fundamental principles of mathematics, computing, and domain-specific disciplines.

PO 3: Design and Development of Solutions

Identify, analyze, design, and evaluate computer-based systems, components, and processes to meet specific application needs, considering public health, safety, cultural, societal, and environmental factors.

PO 4: Professional Ethics

Provide ethical computing solutions, adhering to professional practices and cyber regulations.

PO 5: Individual and Team Work

Work effectively as an individual, as a member, or as a leader in diverse and multidisciplinary teams to develop computing solutions.

PO 6: Innovation and Entrepreneurship

Apply innovative skills and initiate significant opportunities to drive individual and societal progress through entrepreneurship.

Program Specific Outcomes (PSOs):

PSO 1:

Demonstrate global expertise in technological planning and the development of software applications relevant to modern advancements.

PSO 2:

Develop the ability to engage in independent and lifelong learning, continually enhancing skills in computer applications.

PSO 3:

Acquire in-depth knowledge and foster sustained learning, driving innovation, modernization, and research to meet global demands.

Sustainable Development Goal(SDG)- Course Mapping Matrix

Semester-I

Course	SDG 4	SDG 8	SDG 9	SDG 11	SDG 12	SDG 13	Remark
Mathematical Foundations of Computer Science	✓	✓	✓				Analytical and computational thinking
Data Structures using C	✓	✓	✓				Efficient data organization and programming
Object Oriented Programming using Python	✓	✓	✓				Programming skills and software development
Operating Systems	✓	✓	✓				Secure and efficient computing systems
Artificial Intelligence	✓	✓	✓	✓		✓	AI for smart and sustainable solutions
Managerial Economics and Accountancy	✓	✓	✓		✓		Sustainable business and entrepreneurship
Data Structures using C Lab	✓	✓	✓				Practical implementation of data structures
Python Programming Lab	✓	✓	✓				Hands-on application development
Soft Skills Lab	✓	✓					Employability, communication and leadership skills

Notes

SDG 4 (Quality Education) is mapped to every course because all courses contribute directly to knowledge, skills, and lifelong learning.

SDG 8 (Decent Work and Economic Growth) is mapped to almost all MCA courses due to their focus on employability and professional competencies.

SDG 9 (Industry, Innovation and Infrastructure) is central to the MCA curriculum and is therefore associated with nearly all technical courses.

SDG 11 (Sustainable Cities and Communities) is mapped where technologies such as AI, databases, machine learning, and data engineering contribute to smart city and digital governance applications.

SDG 12 (Responsible Consumption and Production) is included for courses involving efficient data management, sustainable business practices, and responsible resource utilization.

SDG 13 (Climate Action) is mapped to AI, Machine Learning, and Internship/Projects where solutions can address environmental sustainability and climate-related challenges.

Sustainable Development Goal (SDG)- Course Mapping Matrix**Semester-II**

Course	SDG 4	SDG 8	SDG 9	SDG 11	SDG 12	SDG 13	Remark
Applied Mathematics	✓	✓	✓				Statistical analysis and decision making
Database Systems	✓	✓	✓	✓			Smart databases and digital governance
Design and Analysis of Algorithms	✓	✓	✓				Efficient algorithmic solutions
Machine Learning	✓	✓	✓	✓	✓		AI for sustainability and intelligent systems
Data Engineering with Python	✓	✓	✓	✓	✓	✓	Data-driven decision support systems
Organization Behaviour	✓	✓	✓		✓		Ethical leadership and sustainable organizations
Database Systems Lab	✓	✓	✓	✓			Practical database application development
Design and Analysis of Algorithms Lab	✓	✓	✓				Algorithm implementation and optimization
Machine Learning Lab	✓	✓		✓	✓	✓	Practical AI for sustainable applications
Summer Internship / Mini Project				✓	✓	✓	Industry collaboration and real-world innovation

Notes

SDG 4(Quality Education) is mapped to every course because all courses contribute directly to knowledge, skills, and lifelong learning.

SDG 8 (Decent Work and Economic Growth) is mapped to almost all MCA courses due to their focus on employability and professional competencies.

SDG 9(Industry, Innovation and Infrastructure) is central to the MCA curriculum and is therefore associated with nearly all technical courses.

SDG 11(Sustainable Cities and Communities) is mapped where technologies such as AI, databases, machine learning, and data engineering contribute to smart city and digital governance applications.

SDG 12(Responsible Consumption and Production) is included for courses involving efficient data management, sustainable business practices, and responsible resource utilization.

SDG 13(Climate Action) is mapped to AI, Machine Learning, and Internship/Projects where solutions can address environmental sustainability and climate-related challenges.

Definition of Credit

Type	Credit
1 Hrs Lecture(L) Per Week	1 Credit
1 Hrs Tutorial (T) Per Week	1 Credit
1 Hrs Practical (P) Per Week	0.5 Credit
2 Hours Practical(P) Per Week	1 Credit

Abbreviation and Definition

Abbreviations	Definitions
PCC	Professional Core Course
PEC	Professional Elective Course
MGC	Management Course
HSC	Humanities Course
LCC	Laboratory Core Course
CIE	Continuous Internal Evaluation
SEE	Semester End Evaluation

Scheme of Examination

	CIE- Marks	SEE-Marks	CIE-Duration	SEE- Durations
Theory	30	70	1 Hr	3 Hr
Practical	25	50	2 Hr	3 Hr
Internship	50	-	3 Hr	3
Project	50	100	3 Hr	3 Hr

Credit Table R26(2026-2028)

Semester	Credits
I	24
II	24
III	24
IV	22
Total	98

MCA Year-I Semester-I

Course Code	Course Title	Nature	Credits	Max Marks (CCE+ESE)
Theory				
OM-PCC101	Mathematical Foundations of Computer	Professional Core Course	3	30+70
OM-PCC102	Data Structures using C	Professional Core Course	3	30+70
OM-PCC103	Object Oriented Programming Using Python	Professional Core Course	3	30+70
OM-PCC104	Operating Systems	Professional Core Course	3	30+70
OM-PCC105	Artificial Intelligence	Professional Core Course	3	30+70
OM-MGC106	Managerial Economics and Accountancy	Management Course	3	30+70
Practicals				
OM-LCC102	Data Structures using C Lab	Laboratory Core Course	2	25+50
OM-LCC103	Python Programming Lab	Laboratory Core Course	2	25+50
OM-HSC110	Soft Skills Lab	Humanities Course	2	25+50
Total			24	255+570

MCA Year-I Semester-II

Course Code	Course Title	Nature	Credits	Max Marks (CCE+ESE)
OM-PCC201	Applied Mathematics	Professional Core Course	3	30+70
OM-PCC202	Database Systems	Professional Core Course	3	30+70
OM-PCC203	Design and Analysis of Algorithms	Professional Core Course	3	30+70
OM-PCC204	Machine Learning	Professional Core Course	3	30+70
OM-PCC205	Introduction to Data Engineering	Professional Core Course	3	30+70
OM-PCC206	Computer Networks	Professional Core Course	3	30+70
Practicals				
OM-LCC202	Database Systems Lab	Laboratory Core Course	2	25+50
OM-LCC203	Design and Analysis of Algorithms Lab	Laboratory Core Course	2	25+50
OM-LCC204	Machine Learning Lab	Laboratory Core Course	2	25+50
OM-SIP310	Summer Internship/ Mini Project	Laboratory Core Course		
Total			24	255+570

Detailed Syllabus for Semester I

OM-PCC101**Mathematical Foundations of Computer Science****Credits: 3****Course Objectives**

1. To learn logic theory and Boolean Algebra related to computer science
2. To understand relations and functions.
3. To gain insights into recurrence relation.
4. To comprehend algebraic structure.
5. To study graph theory and concepts of trees.

Course Outcomes

1. Solve logic problems.
2. Represent the relations and functions.
3. Create recurrence relation.
4. Apply algebraic structures.
5. Work on various graph and tree concepts.

Unit-I

Fundamentals of logic: Basic connective and truth tables, logical equivalence, logical implication, Use of quantifiers, definitions and the proof of theorems. Set theory: Sets and subsets, set operations, and the laws of set theory, counting and Venn diagrams. Properties of the integers: The well-ordering principle, recursive definitions, division algorithms, fundamental theorem of arithmetic.

Unit-II

Relations and functions: Cartesian product, functions onto functions, special functions, pigeon-hole principle, composition and Inverse functions. Relations: Partial orders, equivalence relations and partitions. Principle of inclusion and exclusion: Principles of inclusion and exclusion, generalization of principle.

Unit-III

Generating functions: Introductory examples, definitions and examples, partitions of integers. Recurrence relations: First-order linear recurrence relation, second-order linear homogeneous recurrence relation with constant coefficients.

Unit-IV

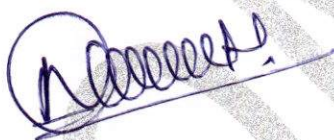
Algebraic structures: Algebraic system general properties, semi groups, monoids, homomorphism, groups, residue arithmetic.

Unit-V

Graph theory: Definitions and examples, subgraphs, complements and graph isomorphism, vertex degree, planar graphs, Hamiltonian parts and cycles. Trees: Definition, properties and examples, rooted trees, spanning trees and minimum spanning trees.

Suggested readings

1. Mott Joe L Mott, Abraham Kandel and Theodore P Baker, Discrete mathematics for computer scientists and mathematicians, Prentice Hall NJ, Second Edition 2015.
2. R. K. Bisht and H.S Dhami, Discrete mathematics, Oxford higher education, 2015.
3. Bhavanari Satyanarayana, Tumurukota Venkata Pradeep Kumar and Shaik Mohiddin Shaw, Mathematical foundations of computer science, BSP, 2016.
4. Ralph P. Grimaldi, Discrete and combinatorial mathematics, 5th edition, Pearson, 2004.



PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-PCC102**Data Structures using C****Credits: 3****Course Objectives**

1. To learn the features of C.
2. To learn the linear and non-linear data structures.
3. To explore the applications of linear and non-linear data structures.
4. To learn to represent data using graph data structures.
5. To learn the basic sorting and searching algorithms.

Course Outcomes

1. Implement linear and non-linear data structure operations using C.
2. Suggest appropriate linear/non-linear data structures for any given data set.
3. Apply hashing concepts to a given problem.
4. Modify or suggest a new data structure for an application.
5. Appropriately choose the sorting algorithm for an application.

Unit-I

Structure of a C program - compilation and linking processes, Constants, Variables, Data Types, Expressions using operators in C, Managing Input and Output operations, Decision Making and Branching, Looping statements. Arrays, Initialization, Declaration, One, dimensional and Two, dimensional arrays. Strings, String operations, String Arrays. Simple programs, sorting, searching, matrix operations.

Unit-II

Functions: Pass by value, Pass by reference, Recursion, Pointers, Definition, Initialization, Pointer arithmetic. Structures and unions: definition, Structure within a structure, Union, Programs using structures and unions, Storage classes, Preprocessor directives.

Unit-III

Arrays and their representations. Stacks and Queues, Applications. Linked lists, Single, circular, and doubly linked lists, Applications.

Unit-IV

Trees, Binary Trees, Binary tree representation and traversals, Applications of trees. Binary Search Trees, AVL trees. Graph and its representations, Graph Traversals.

Unit-V

Linear Search, Binary Search. Sorting: Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort. Hashing, Types of Hashing, Collision resolution techniques.

Suggested readings

1. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. Fundamentals of Data Structures in C. 2nd ed., Universities Press (India) Pvt. Ltd., 2008.
2. Dey, Pradip, and Manas Ghosh. Programming in C. 2nd ed., Oxford University Press, 2013.
3. Weiss, Mark Allen. Data Structures and Algorithm Analysis in C. 2nd ed., Pearson Education, 1997.



PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-PCC103**Object Oriented Programming using Python****Credits: 3**

Course Objectives: After completing this course, students will be able to:

1. Understand Python programming fundamentals, including data types, operators, input/output operations, control structures, and object-oriented programming concepts.
2. Apply object-oriented programming principles such as classes, objects, inheritance, polymorphism, abstraction, and encapsulation in Python applications.
3. Develop modular Python programs using strings, functions, recursion, and different parameter-passing techniques.
4. Implement exception handling mechanisms to develop robust and reliable Python applications.
5. Design graphical user interfaces and interactive applications using Tkinter, Custom Tkinter, and Streamlit frameworks.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Explain Python programming concepts, data types, operators, control statements, and object-oriented programming principles.
2. Develop Python programs using classes, constructors, methods, inheritance, polymorphism, abstract classes, and interfaces.
3. Create modular and reusable applications using strings, functions, recursion, and various argument-passing mechanisms.
4. Handle runtime errors effectively using exception handling techniques, assertions, and user-defined exceptions.
5. Design and develop GUI-based desktop and web applications using Tkinter, Custom Tkinter, and Streamlit for interactive software solutions.

Course Contents**Unit-I**

Introduction to Python: Features of Python, Data types, Operators, Input and output, Control Statements.
Introduction to Object Oriented Concepts: Features of Object-oriented programming system (OOPS) -Classes and Objects, Encapsulation, Abstraction, Inheritance, Polymorphism.

Unit-II

Classes and Objects: Creating a class, The Self variable, Constructor, Types of Variables, Namespaces, Types of Methods, Inheritance and Polymorphism - Constructors in inheritance, the super() method, types of inheritance, polymorphism, abstract classes and interfaces.

Unit-III

Strings: Creating strings and basic operations on strings, string testing methods. Functions: Defining a function, calling a function, returning multiple values from a function, functions are first class objects, formal and actual arguments, positional arguments, recursive functions.

Unit-IV

Exception: Errors in a Python program, exceptions, exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions.

Unit-V

Introduction to GUI Programming with Tkinter: Event Handling, Layout Managers (Pack, Grid, Place), Modern Tkinter using Custom Tkinter, Streamlit for Rapid GUI Development.

Suggested Readings

1. Rao, R. Nageswara. Core Python Programming. 3rd ed., Dreamtech Press, 2021.
2. Philips, Dusty. Python 3 Object-Oriented Programming. 4th ed., Packt Publishing, 2021.

P.V. Sridhar

PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.

Ajay

Indray

Ramesh

OM-PCC104**Operating Systems****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the fundamental concepts of operating systems, process management, CPU scheduling, synchronization, and deadlock handling mechanisms.
2. Analyze memory management and virtual memory techniques for efficient utilization of system resources.
3. Explore file systems, storage management, disk scheduling, and I/O system operations in modern operating systems.
4. Understand operating system security, protection mechanisms, access control models, and threat mitigation techniques.
5. Examine the architecture and functionalities of modern operating systems such as Linux and Windows.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Explain operating system services, process management concepts, multithreading models, CPU scheduling algorithms, synchronization techniques, and deadlock management methods.
2. Apply memory management techniques including paging, swapping, virtual memory, page replacement algorithms, and thrashing control mechanisms.
3. Analyze file system structures, storage management, disk scheduling algorithms, I/O systems, and file allocation methods.
4. Evaluate security and protection mechanisms including authentication, access control, protection domains, and security threat management.
5. Demonstrate understanding of Linux and Windows operating system architectures, process management, memory management, file systems, networking, and system services.

Unit-I

Introduction to Operating Systems: OS services, process concepts, process scheduling, multithreaded programming, multithreaded models, CPU scheduling algorithms, the critical section problem, semaphores, monitors, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection.

Unit-II

Memory Management: contiguous memory allocation, paging, structure of the page table, swapping. Virtual Memory Management: Demand paging, page replacement, thrashing.

Unit-III

Mass-storage structure: overview of mass storage structure, HDD scheduling, I/O systems: I/O systems overview, transforming I/O requests to hardware operations. File System Interface: File concept, access methods, and protection. File System Implementation: File system structure, directory implementation, allocation methods.

Unit-IV

Security and Protection: The security problem, program threats, system and network threats, user authentication. Protection: Goals of protection, principles of protection, domains of protection, access matrix, mandatory access control, role-based access control.

Unit-V

Case Studies: The Linux System-Design principles, Kernel modules, Process management, Scheduling, Memory management, File systems, Input and Output, Inter process. communication. Windows 10-Design principles, System components. Terminal services and fast user switching File systems, Networking, Programmer interface.

Suggested readings:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. 10th ed., Wiley, 2018.
2. Stallings, William. Operating Systems: Internals and Design Principles. 9th ed., Pearson, 2021.
3. Tanenbaum, Andrew S., and Herbert Bos. Modern Operating Systems. 5th ed., Pearson, 2022.

P.V. Sudha

PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.

Aravind

Ankur

Ramesh

OM-PCC105**Artificial Intelligence****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the foundations of Artificial Intelligence, intelligent agents, rationality, and the characteristics of intelligent environments.
2. Apply search techniques, optimization methods, constraint satisfaction approaches, and adversarial search strategies to solve AI problems.
3. Analyze logical reasoning, knowledge representation, and inference mechanisms using propositional and first-order logic.
4. Explore probabilistic reasoning, Bayesian networks, and hidden Markov models for decision-making under uncertainty.
5. Understand decision-theoretic approaches, Markov decision processes, POMDPs, and multi agent systems for intelligent decision making.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Explain the fundamental concepts of Artificial Intelligence, intelligent agents, rational behavior, and environmental characteristics in AI systems.
2. Implement and analyze uninformed search, heuristic search, optimization methods, constraint satisfaction techniques, and adversarial search algorithms such as Minimax and Alpha-Beta Pruning.
3. Apply propositional logic, first-order logic, forward chaining, backward chaining, resolution, and knowledge representation techniques to AI problems.
4. Analyze uncertainty using probability theory, Bayes' rule, Naïve Bayes models, Bayesian networks, and Hidden Markov Models.
5. Evaluate decision-making techniques using decision networks, value of information, MDPs, POMDPs, and multi agent system concepts for intelligent problem solving.

Unit-I

Introduction: The foundations of AI, History of AI, the state of the art, Intelligent agents: agents and environment, good behaviour: the concept of rationality, the nature of environments, the structure of agents.

Unit-II

Solving problems by searching: problem solving agents, search algorithms, uninformed search strategies, informed or heuristic search strategies, heuristic functions. Search in Complex environments: local search and

optimization problems, Constraint Satisfaction problems: defining constraint satisfaction problem, Adversarial Search and Games: the minimax search algorithm, alpha-beta pruning.

Unit-III

Logical Agents: propositional logic, propositional theorem proving, First Order Logic: Syntax and Semantics of first order logic, Inference in First Order Logic: Forward Chaining, Backward Chaining, Resolution, Knowledge Representation: Ontological Engineering, Categories and Objects.

Unit-IV

Quantifying uncertainty: acting under uncertainty, basic probability notation, Bayes' rule and its use, Naïve Bayes models, Probabilistic reasoning: the semantics of Bayesian networks, exact inference in Bayesian network, Probabilistic reasoning over time: time and uncertainty, hidden Markov models.

Unit-V

Making Simple Decisions: decision networks, the value of information, Making complex decisions: algorithms for MDPs bandit problems, algorithms for solving POMDPs, Multiagent decision making: properties of multi agent environments.

Suggested Readings

1. Stuart J. Russell, Peter Norvig, Artificial Intelligence - A Modern Approach, Pearson Education, 4th Edition, 2021
2. Elaine Rich, Kevin Knight, Shivashankar B. Nair, Artificial Intelligence, Third Edition,
3. TataMcGraw-Hill, 2008.
4. Dheepak Khemani, "A First Course in Artificial Intelligence", McGraw-Hill, 2013.

NPTEL Course

1. NPTEL Artificial Intelligence Course by Prof. Dasgupta: <https://nptel.ac.in/courses/106105079>

Web Links

2. <https://cloud.google.com/dialogflow>
3. <https://cloud.google.com/community/tutorials/kubernetes-ml-ops>
4. <https://www.tensorflow.org/tutorials/images/cnn>

OM-MGC106**Managerial Economics and Accountancy****Credits: 3****Course Objectives**

1. To learn important concepts of Managerial Economics and apply them to evaluate business decisions.
2. To understand various parameters that determine consumers' behavior.
3. To evaluate the factors that affect production.
4. To understand the concepts of capital budgeting and payback period.
5. To study the concepts of various bookkeeping methods.

Course Outcomes

1. Apply the fundamental concepts of managerial economics to evaluate business decisions and
2. understand types of demand and factors related to it.
3. Identify different types of markets and determine price-output under perfect competition.
4. Determine working capital requirements and payback.
5. Analyze and interpret financial statements through ratios.
6. Determine double entry bookkeeping methods

Unit-I

Meaning and Nature of Managerial Economics: Managerial Economics and its usefulness to engineers, Fundamental Concepts of Managerial Economics - Scarcity, Marginalism, Equi- marginalism, Opportunity costs, Discounting, Time Perspective, Risk and Uncertainty, Profits, Case study method.

Unit-II

Law of Demand and Supply: Law of Demand, Determinants, Types of Demand; Elasticity of Demand (Price, Income, and Cross-Elasticity), Demand Forecasting, Law of Supply and Concept of Equilibrium.

Unit-III

Theory of Production and Markets: Production Function, Law of Variable Proportion, ISO quants, Economies of Scale, Cost of Production (Types and their measurement), Concept of Opportunity Cost, Concept of Revenue, Cost-Output relationship, Break-Even Analysis, Price-Output determination under Perfect Competition and Monopoly.

Unit-IV

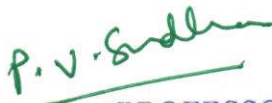
Working Capital Management and Capital Budgeting: Concepts, Significance, determination, and estimation of fixed and variable working capital requirements, sources of capital. Introduction to capital budgeting methods - traditional and modern methods with problems.

Unit-V

Accounting: Meaning, Significance, Principles of double-entry bookkeeping, Journal, Ledger accounts, Subsidiary books, Trial Balance, preparation of Final Accounts with simple adjustments, Analysis and interpretation of Financial Statements through Ratios.

Suggested readings

1. Mehta P.L., Managerial Economics - Analysis, Problems and Cases, Sultan Chand & Sons
2. Educational Publishers, 2011.
3. Maheswari S.N., Introduction to Accountancy, Vikas Publishing House, 2005.
4. Pandey I.M., Financial Management, Vikas Publishing House, 2009.
5. S.P. Jain and K.L. Narang, Financial Accounting, Kalyan Publishers, 2018.
6. M. Hanif and A. Mukherjee, Modern Accountancy, McGraw Hill, 3rd Edition, 2018.


PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.







OM-LCC102**Data Structures using C Lab****Credits: 2**

Course Objectives: After completing this laboratory course, students will be able to:

1. Apply C programming constructs such as control statements, functions, pointers, structures, and unions to solve computational problems.
2. Develop programs using arrays, strings, recursion, and dynamic memory concepts for efficient data processing.
3. Implement fundamental linear and non-linear data structures using C programming.
4. Design and analyze searching, sorting, and graph traversal algorithms using appropriate data structures.
5. Evaluate the performance of different data organization and retrieval techniques including trees, graphs, and hashing methods.

Course Outcomes: Upon successful completion of the laboratory course, students will be able to:

1. Develop C programs using operators, control structures, functions, pointers, structures, unions, and recursion for problem solving.
2. Implement array, matrix, and string manipulation programs and demonstrate dynamic memory access techniques.
3. Create and manipulate linear data structures such as stacks, queues, singly linked lists, circular linked lists, and doubly linked lists.
4. Implement non-linear data structures including binary trees, binary search trees, and graph representations with traversal algorithms such as BFS and DFS.
5. Analyze and compare the performance of searching, sorting, and hashing techniques using suitable collision resolution and optimization methods.

Lab Experiments

1. Write a C program to perform arithmetic operations, decision-making statements, and looping constructs to solve simple computational problems.
2. Develop a C program to perform operations on one-dimensional and two-dimensional arrays, including matrix addition, subtraction, and multiplication.
3. Write a C program to perform string operations such as string length, concatenation, comparison, reversal, and palindrome checking.
4. Implement programs using functions to demonstrate pass-by-value, pass-by-reference, and recursion (factorial and Fibonacci series).
5. Develop a C program illustrating pointer operations, pointer arithmetic, and dynamic memory access.
6. Write a C program using structures and unions to maintain and display student or employee records.

7. Implement Stack operations (Push, Pop, Peek) using arrays and demonstrate applications such as expression evaluation.
8. Implement Queue operations (Enqueue, Dequeue, Display) using arrays and demonstrate circular queue implementation.
9. Develop programs to create and manipulate:
 10. Singly Linked List
 11. Circular Linked List
 12. Doubly Linked List
13. including insertion, deletion, and traversal operations.
14. Implement Binary Tree creation and perform Preorder, Inorder, and Postorder traversals.
15. Develop a program to create a Binary Search Tree (BST) and perform insertion, deletion, and searching operations.
16. Implement Graph representation using adjacency matrix/adjacency list and perform Breadth First Search (BFS) and Depth First Search (DFS) traversals.
17. Implement Linear Search and Binary Search algorithms and compare their performance for different input sizes.
18. Develop programs for the following sorting techniques and compare their execution times:
 - Selection Sort
19. . Bubble Sort
 - Insertion Sort
 - Merge Sort
20. . Quick Sort
21. Implement a Hash Table with collision resolution techniques such as:
 - Linear Probing
 - Quadratic Probing
 - Chaining
22. and analyse their performance.

Mini Project

Student Record Management System Using Data Structures

Develop a menu-driven application that performs:

- . Student record creation, update, deletion, and search
- . Dynamic storage using linked lists
- . Sorting student records using different sorting algorithms
- . Fast retrieval using hashing
- . Tree-based searching using BST

Tools: GCC Compiler, Code :: Blocks, Dev-C++, Turbo C (or any modern C IDE)

OM-LCC103**Python Programming Lab****Credits: 2**

Course Objectives: After completing this laboratory course, students will be able to:

1. Apply Python programming fundamentals and object-oriented programming concepts to develop structured software solutions.
2. Design and implement Python applications using classes, inheritance, polymorphism, abstraction, and modular programming techniques.
3. Develop programs using strings, functions, recursion, and advanced parameter-passing mechanisms.
4. Implement robust applications through exception handling, assertions, and user-defined exceptions.
5. Design and develop interactive desktop and web-based applications using Tkinter, CustomTkinter, and Streamlit frameworks.

Course Outcomes: Upon successful completion of the laboratory course, students will be able to:

1. Develop Python programs utilizing operators, control structures, loops, and object-oriented programming concepts to solve computational problems.
2. Create reusable and maintainable applications using classes, constructors, inheritance hierarchies, polymorphism, abstract classes, and different method types.
3. Apply string processing, functions, recursion, and argument-passing techniques to build modular and efficient Python programs.
4. Handle runtime errors effectively using exception handling mechanisms, assertions, and custom exceptions for reliable software development.
5. Design and implement GUI-based desktop and web applications using Tkinter, CustomTkinter, and Streamlit, including a complete student management system with file handling and user-friendly interfaces.

Lab Experiments

1. Write a Python program to perform arithmetic, relational, logical, and bitwise operations on user-input values and display the results.
2. Develop a Python program using conditional statements and loops to generate patterns, check prime numbers, and compute factorials.
3. Create a class Student and demonstrate Object-Oriented Programming concepts such as classes, objects, encapsulation, abstraction, inheritance, and polymorphism.
4. Write a Python program to create a class with constructors, instance variables, class variables, and different types of methods (instance, class, and static methods).
5. Develop a Python program demonstrating single, multilevel, multiple, hierarchical, and hybrid inheritance.

6. Create a Python program to demonstrate method overriding, the super() method, and abstract classes using the abc module.
7. Write a Python program to perform string operations including concatenation, slicing, searching, replacing, splitting, and string testing methods.
8. Develop Python functions to demonstrate positional arguments, keyword arguments, default arguments, variable-length arguments, and returning multiple values.
9. Implement recursive functions to solve problems such as factorial calculation, Fibonacci series generation, and Tower of Hanoi.
10. Write a Python program to handle multiple exceptions such as ZeroDivisionError, ValueError, IndexError, and FileNotFoundError.
11. Develop a Python program that uses the assert statement to validate user input and handle assertion failures.
12. Create a user-defined exception and demonstrate raising and handling custom exceptions in a real-world scenario such as banking or student management.
13. Design a Tkinter GUI application for a Student Registration Form using Labels, Entry widgets, Buttons, and Layout Managers (Pack, Grid, and Place).
14. Develop a modern GUI application using CustomTkinter to create a Calculator or Login System with event handling.
15. Create a Streamlit web application that accepts user input through forms and displays processed results dynamically.

Mini Project

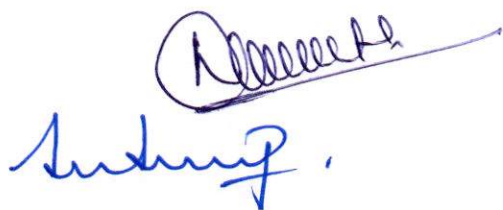
Interactive Student Management System

Develop a GUI-based application using Tkinter / Custom Tkinter or Streamlit that allows users to:

- Add, update, delete, and search student records.
- Validate user input using exception handling.
- Implement OOP concepts such as classes, inheritance, and polymorphism.
- Store and retrieve data using files.
- Provide a user-friendly desktop or web interface.

Tools

- Python 3.x
- Tkinter
- CustomTkinter
- Streamlit
- VS Code / PyCharm / Jupyter Notebook



PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-HSC110**Soft Skills Lab****Credits: 2****Course Objectives**

1. Learn conversational skills.
2. Learn reading strategies.
3. Learn time management.
4. Learn stress management.
5. Learn career planning.

Course Outcomes

1. Express conversational skills.
2. Specify reading strategies.
3. Perform time management.
4. Perform stress management.
5. Explore career planning.

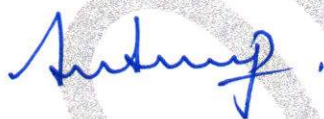
Lab Activities

1. Conversation skills, listening dialogues from TV/radio/Ted Talks/Podcasts.
2. Group discussion.
3. Interview Skills, Making Presentation.
4. listening to lectures and News programmes, listening to talk shows.
5. Watching videos on interesting events on YouTube,
6. Reading different genres of texts ranging from newspapers to philosophical treatises.
7. Reading strategies - graphic organizers, Reading strategies - summarizing
8. Reading strategies - interpretation reports.
9. Writing cover letters, resumes.
10. Writing for publications, letters, memos, emails, and blogs.
11. Civil service (language-related), verbal ability.
12. Motivation, self-image.
13. Goal setting, managing changes.
14. Time management, stress management.
15. Leadership traits.
16. Teamwork.

17. Career and life planning.
18. Multiple intelligences.
19. Emotional intelligence.
20. Spiritual quotient (ethics).
21. Intercultural communication.
22. Creative and critical thinking.
23. Learning styles and strategies.

Suggested readings

1. Business English Certificate (BEC): Official Preparation Materials. Cambridge University Press & Assessment, 2023.
2. Graded Examinations in Spoken English (GESE) and Spoken English for Work: Official Preparation Materials. Trinity College London, 2023.
3. Cambridge IELTS Practice Tests. Cambridge University Press & Assessment, 2023.
4. Interactive Multimedia Programs on Managing Time and Stress. Tata Interactive Systems, 2018. (Representative multimedia learning resource.)
5. Personality Development. CD-ROM, Times Multimedia, Mumbai, 2005. (Legacy multimedia resource; no newer edition is available.)
6. Sherfield, Robert M., Rhonda J. Montgomery, and Patricia G. Moody. Developing Soft Skills. 5th ed., Pearson, 2014.



PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



SEMESTER-II

Detailed Syllabus for Semester II

OM-PCC201**Applied Mathematics****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the concepts of partial differentiation and functions of several variables.
2. Apply total differentiation and optimization techniques for solving problems involving maxima and minima.
3. Analyze systems of linear equations and determine linear dependence and independence of vectors and matrices.
4. Understand vector spaces, subspaces, bases, eigen values, and eigenvectors in linear algebra.
5. Apply diagonalization techniques and use eigen value methods in solving differential equations and related applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Compute partial derivatives and analyze continuity and limits of functions of two variables.
2. Solve problems involving total differentials, implicit functions, and constrained optimization using Lagrange multipliers.
3. Solve systems of linear equations and determine the linear independence or dependence of vectors and matrices.
4. Demonstrate knowledge of vector spaces, bases, eigen values, eigenvectors, and characteristic equations.
5. Diagonalize matrices and apply eigen value techniques to solve differential equations and engineering applications.

UNIT - I

Partial Differentiation Introduction, Functions of Two Variables, Neighbourhood of a Point $((a, b))$, Continuity of a Function of Two Variables, Continuity at a Point, Limit of a Function of Two Variables, Partial Derivatives, Homogeneous Functions

UNIT - II:

Total Differentiation and Optimization Theorem on Total Differentials, Composite Functions, Differentiation of Composite Functions, Implicit Functions, Maxima and Minima of Functions of Two Variables, Lagrange's Method of Undetermined Multipliers.

UNIT - III:

Linear Equations in Linear Algebra Systems of Linear Equations, Consistent and Inconsistent Systems, Solution Sets of Linear Systems, Trivial and Non-Trivial Solutions, Linear Independence, Linear Independence of Matrix Columns, Characterization of Linearly Dependent Sets

UNIT -IV:

Vector Spaces and Eigen values Vector Spaces and Subspaces, Linearly Independent Sets, Bases, Eigenvalues and Eigenvectors, The Characteristic Equation, Diagonalizing Matrices with Non-Distinct Eigenvalues, Applications to Differential Equations

UNIT- V:

Probability& Sampling Basic terminology, three types of probability, probability rules, statistical independence, statistical dependency, Bayes' theorem. Probability Distributions - Random variables, expected values, binomial distribution, Poisson distribution, normal distribution, choosing the correct distribution. Sampling and Sampling Distributions - Random sampling, Non-Random Sampling distributions, operational considerations in sampling. Estimation - Point estimates, interval estimates, confidence intervals, calculating interval estimates of the mean and proportion, t-distribution, determination of sample size in estimation.

Suggested Readings

1. Lay, David C. Linear Algebra and Its Applications. 6th ed., Pearson Education Limited, 2022.
2. Lang, Serge. Introduction to Linear Algebra. 3rd ed., Springer, 2000.
3. Strang, Gilbert. Linear Algebra and Its Applications. 4th ed., Cengage Learning, 2005.
4. T. Veerarajan, Probability, Statistics and Random Processes, 3rd Edition, Tata McGraw-Hill, 2008



P. V. Sudha

PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-PCC202**Database Systems****Credits: 3****Course Objectives**

1. Introduce database concepts along with ER modeling.
2. Learn about relational databases and SQL query language.
3. Define advanced SQL.
4. Study database transactions and explore concurrency concepts.
5. Introduce NoSQL.

Course Outcomes

1. Explain database concepts and model requirements using the ER model.
2. Formulate relational algebra queries from textual specifications.
3. Write SQL queries for given problems.
4. Elaborate on indexing and hashing, and describe concurrency control concepts.
5. Understand NoSQL technology.

Unit-I

Introduction: Database system applications, purpose of database systems, views of values, nested sub-queries, complex queries, views, modification of the database, joined relations data, database languages, relational databases, database design, object-based and semi-structured databases, data storage and querying, transaction management, data mining and analysis, database architecture, database users and administrators. Database Design and the E-R Model: Overview of the design process, the entity-relationship model, constraints, entity-relationship diagrams, entity-relationship design issues, weak entity sets, extended E-R features, database design for a banking enterprise, reduction to relational schemas, and other aspects of database design.

Unit-II

Relational Model: Structure of relational databases, fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification of databases. Structured Query Language (SQL): Data definition, basic structure of SQL queries, set operations, aggregate functions, null.

Unit-III

Advanced SQL: SQL data types and schemas, integrity constraints, authorization, embedded SQL, dynamic SQL, functions and procedural constructs, recursive queries, and advanced SQL features. Relational Database Design: Features of good relational design, atomic domains and first normal form, functional dependency theory, decomposition using functional dependencies.

Unit-IV

Indexing and Hashing: Basic concepts, ordered indices, B+tree index files, Btree index files, multiple- key access, static hashing, dynamic hashing, comparison of ordered indexing and hashing, bitmap indices. Index Definition in SQL Transactions: transaction concepts, transaction states, implementation of atomicity and durability, concurrent executions, serializability, recoverability, implementation of isolation, testing for serializability.

Unit-V

Concurrency Control: Lock-based protocols, timestamp-based protocols, validation-based protocols, multiple granularity, multi-version schemes, deadlock handling, insert and delete operations, weak levels of consistency, concurrency of index structures. Recovery System: Failure classification, storage structure, recovery and atomicity, log-based recovery, recovery with concurrent transactions, buffer management, failure with loss of nonvolatile storage, advanced recovery techniques, remote backup systems. NoSQL: Need for NoSQL, aggregate data models, more details on data models, distribution models, consistency, version stamps, map-reduce, key-value databases, document databases, column- family stores, graph databases, schema migrations.

Suggested readings

1. Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. Database System Concepts. 7th ed., McGraw Hill, 2020.
2. Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. 3rd ed., McGraw-Hill Education, 2003.
3. Elmasri, Ramez, Shamkant B. Navathe, and Shyam K. Gupta. Fundamentals of Database Systems. 7th ed., Pearson, 2017.
4. Tiwari, Shashank. Professional NoSQL. 1st ed., Wiley, 2011.




PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.




OM-PCC203**Design and Analysis of Algorithms****Credits: 3****Course Objectives**

1. Learn algorithms time complexity.
2. Learn divide and conquer approach.
3. Learn greedy method.
4. Learn dynamic programming.
5. Learn backtracking.

Course Outcomes

1. Carry out algorithms time complexity.
2. Explain divide and conquer approach.
3. Illustrate greedy method.
4. Elaborate dynamic programming.
5. Explore backtracking.

Unit-I

Introduction to Algorithms: Algorithm Specifications, Performance Analysis, Randomized Algorithms. Elementary Data Structures: Stacks and Queues, Trees, Dictionaries, Priority Queues, Sets and Disjoint Set Union, Graphs.

Unit-II

Divide and Conquer: Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Selection sort, Strassen's Matrix Multiplication, Convex Hull. The Greedy Method: Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees, Single Source Shortest Paths.

Unit-III

Dynamic Programming: General Method, Multistage Graphs, All-Pairs Shortest Paths, Single-Source Shortest Paths, Optimal Binary Search Trees, 0/1 Knapsack, The Traveling Salesperson Problem. Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected Components and Spanning Trees, Biconnected Components and DFS.

Unit-IV

Back Tracking: General Method, 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem. Branch-Bound: The Method, 0/1 Knapsack Problem, Traveling Sales Person.

Unit-V

NP-Hard and NP-Completed Problems: Basic Concepts, Cook's Theorem, NP-Hard Graph Problems, NP-Hard Scheduling Problems, NP-Hard Code Generation, Some Simplified NP-Hard Problems.

Suggested Readings

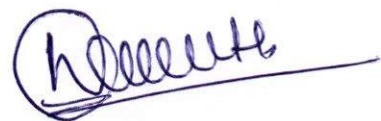
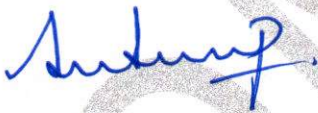
1. Horowitz, Ellis, Sartaj Sahni, and Sanguthevar Rajasekaran. Fundamentals of Computer Algorithms. 2nd ed., Universities Press, 2008.
2. Pannerselvam, R. Design and Analysis of Algorithms. 3rd ed., PHI Learning Pvt. Ltd., 2019.
Pandey, Hari Mohan. Design Analysis and Algorithms. 1st ed., University Science Press, 2009.
3. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. 4th ed., MIT Press, 2022.



P.V. Sridhar

PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-PCC204**Machine Learning****Credits: 3**

Course Objectives: After completing this course, students will be able to:

1. Understand the machine learning lifecycle, problem formulation, data preparation, model development, and evaluation processes.
2. Apply supervised learning techniques for regression and classification tasks using appropriate performance measures.
3. Analyze decision tree and ensemble learning methods for building robust predictive models.
4. Explore dimensionality reduction and unsupervised learning techniques for data analysis and pattern discovery.
5. Develop and evaluate artificial neural network models using Scikit-learn and PyTorch for real-world machine learning applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Explain machine learning concepts, types of learning systems, challenges in machine learning, and the end-to-end machine learning workflow.
2. Implement and evaluate classification and regression models including Linear Regression, Logistic Regression, and multiclass classification techniques using appropriate performance metrics.
3. Develop predictive models using Decision Trees, Random Forests, Boosting, Bagging, Voting Classifiers, and other ensemble learning methods.
4. Apply dimensionality reduction techniques such as PCA and implement clustering and Gaussian Mixture Models for unsupervised learning tasks.
5. Design, train, and evaluate Artificial Neural Networks and Multi-Layer Perceptrons using Scikit-learn and PyTorch, including image classification and deep learning applications.

Unit-I

Machine Learning Landscape: machine learning introduction, use of machine learning, types of machine learning systems, main challenges of machine learning, testing and validating, End-to-End Machine Learning: Working with real data, frame the problem, performance measure, get the data, explore and visualize the data to gain insights, prepare the data for machine learning algorithms, select and train a model, fine-tune your model.

Unit-II

Training a binary classifier: performance measures, multiclass classification, error analysis, multi - label classification, multi-output classification, MNIST, Training Models: Linear regression, Gradient descent, regularized linear models, logistic regression.

Unit-III

Decision Trees: Training and visualizing a decision tree, making predictions, estimating class probabilities, the CART training algorithm, Gini impurity, Entropy, regularization hyperparameters, Ensemble Learning and Random Forest: voting classifiers, bagging and pasting, random forests, boosting, stacking.

Unit-IV

Dimensionality Reduction: the curse of dimensionality, main approaches for dimensionality reduction, PCA, Unsupervised Learning Techniques: clustering algorithms, Gaussian mixtures-using gaussian mixtures for anomaly detection, selecting the number of clustering, Bayesian gaussian mixture models.

Unit-V

Introduction to Artificial Neural Networks: from biological to artificial neurons, building and training MLPs with scikit-learn, hyperparameter tuning guidelines, Building neural networks with PyTorch: PyTorch fundamentals, implementing linear regression, implementing a regression MLP, implementing mini-batch gradient descent using dataloaders, model evaluation, building an image classifier with PyTorch.

Suggested Readings

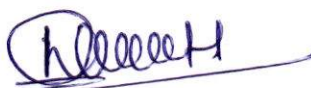
1. Aurelien Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition, O'reilly, 2025.
2. Stephen Marsland, "Machine Learning: An Algorithmic Perspective", Chapman & Hall/CRC, 2nd Edition, 2014.
3. Kevin Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012
4. Ethem Alpaydin, "Introduction to Machine Learning", Third Edition, Adaptive Computation and Machine Learning Series, MIT Press, 2014
5. Tom M Mitchell, "Machine Learning", McGraw Hill Education, 2013.


P.V. Sudha

PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.







OM-PCC205**Fundamentals of Data Engineering****Credits: 3**

Course Objectives: After completing this course, students will be able to:

1. Understand the Role of Data Engineering in Modern Organizations
2. Design Scalable and Reliable Data Architectures.
3. Select and Implement Appropriate Technologies Across the Data Engineering Lifecycle
4. Optimize Data Storage and Ingestion Techniques.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Demonstrate a comprehensive understanding of the data engineering lifecycle, encompassing essential skills, activities, and the roles of a data engineer in modern business settings.
2. Design and assess scalable and reliable data architectures that integrate modern data platforms like data lakes and warehouses.
3. Utilize decision-making skills to select optimal technologies and frameworks based on team capabilities, performance, cost, and interoperability across the data engineering lifecycle.
4. Implement effective data storage solutions by applying knowledge of distributed storage, partitioning, data retention, and modern storage architectures such as data lakehouses and stream-m to-batch systems.
5. Build and manage data ingestion pipelines that handle both bounded and unbounded data while ensuring scalability, data integrity, and reliability through techniques like ETL, change data capture, and message queues.

Unit-I

Data Engineering Described : Definition, Lifecycle, Evolution of the Data Engineer, Data Engineering and Data Science , Data Engineering Skills and Activities, Data Maturity and the Data Engineer, The Background and Skills of a Data Engineer, Business Responsibilities, Technical Responsibilities, The Continuum of Data Engineering Roles, from A to B, Data Engineers Inside an Organization, Internal-Facing Versus External-Facing Data Engineers, Data Engineers and Other Technical Roles, Data Engineers and Business Leadership.

The Data Engineering Lifecycle: The Data Lifecycle Versus the Data Engineering Lifecycle, Generation: Source Systems, Storage, Ingestion, Transformation, Serving Data, Major Undercurrents Across the Data Engineering Lifecycle, Security, Data Management, DataOps, Data Architecture, Orchestration, Software Engineering.

Unit-II

Designing Good Data Architecture: What Is Data Architecture, Enterprise Architecture Defined, Data Architecture Defined, "Good" Data Architecture, Principles of Good Data Architecture, Major Architecture Concepts.

Domains and Services, Distributed Systems, Scalability, and Designing for Failure, Tight Versus Loose Coupling: Tiers, Monoliths, and Microservices, User Access: Single Versus Multitenant, Event-Driven Architecture,

Brownfield Versus Greenfield Projects, Examples and Types of Data Architecture, Data Warehouse, Data Lake, Convergence, Next-Generation Data Lakes, and the Data Platform, Modern Data Stack, Lambda Architecture, Kappa Architecture, The Dataflow Model and Unified Batch and Streaming, Architecture for IoT, Data Mesh, Other Data Architecture Examples, Who's Involved with Designing a Data Architecture.

Unit-III

Choosing Technologies Across the Data Engineering Lifecycle Team Size and Capabilities, Speed to Market, Interoperability, Cost Optimization and Business Value, Total Cost of Ownership, Total Opportunity Cost of Ownership, FinOps, Optimization, Performance, and the Benchmark Wars.

Data Generation in Source Systems: Sources of Data: How Is Data Created? Source Systems: Main Ideas, Files and Unstructured Data, APIs, Application Databases (OLTP Systems), Online Analytical Processing System, Change Data Capture, Logs, Database Logs, CRUD, Insert-Only.

Messages and Streams, Types of Time, Source System Practical Details, Databases, APIs, Data Sharing, Third-Party Data Sources, Message Queues and Event-Streaming Platforms, Undercurrents and Their Impact on Source Systems, Security, Data Management, DataOps, Data Architecture, Orchestration, Software Engineering.

Unit-IV

Storage: Data Storage Systems, Single Machine Versus Distributed Storage, Eventual Versus Strong Consistency, Cache and Memory-Based Storage Systems, The Hadoop Distributed File System, Streaming Storage, Indexes, Partitioning, and Clustering, Data Engineering Storage Abstractions, The Data Warehouse, The Data Lake, The Data Lakehouse, Data Platforms.

Stream-to-Batch Storage Architecture, Big Ideas and Trends in Storage, Data Catalog, Data Sharing, Schema, Separation of Compute from Storage, Data Storage Lifecycle and Data Retention, Single- Tenant Versus Multitenant Storage

Unit-V

Ingestion: What Is Data Ingestion, Key Engineering Considerations for the Ingestion Phase, Bounded Versus Unbounded Data, Frequency, Synchronous Versus Asynchronous Ingestion, Serialization and Deserialization, Throughput and Scalability, Reliability and Durability, Payload, Push Versus Pull Versus Poll Patterns, Batch Ingestion Considerations, Snapshot or Differential Extraction, File-Based Export and Ingestion, ETL Versus ELT

Inserts, Updates, and Batch Size, Data Migration, Message and Stream Ingestion Considerations, Schema Evolution, Late-Arriving Data, Ordering and Multiple Delivery, Replay, Time to Live, Message Size, Error Handling and Dead-Letter Queues, Consumer Pull and Push, Location, Ways to Ingest Data, Direct Database Connection, Change Data Capture, APIs, Message Queues and Event-Streaming Platforms, Managed Data Connectors, Moving Data with Object Storage, EDI, Databases and File Export.

Suggested Readings

1. Joe Reis and Matt Housley, Fundamentals of Data Engineering Plan and Build Robust Data Systems, Published by O'Reilly Media, Inc., July 2022: First Edition.
2. Paul Crickard, Data Engineering with Python: Work with massive datasets to design data models and automate data pipelines using Python, Kindle Edition Packt Publishers.



PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-PCC206**COMPUTER NETWORKS****Credits 03****Course Objectives:**

1. To study the design issues in network layer and various routing algorithms.
2. To introduce internet routing architecture and protocols.
3. To learn the flow control and congestion control algorithms in Transport Layer.
4. To introduce the TCP/IP suite of protocols and the networked applications supported by it.
5. To learn basic and advanced socket system calls.

Course Outcomes:

At the end of the course the student will be able to:

1. Apply the function of each layer of OSI and trace the flow of information from one node to another node in the network.
2. Understand the principles of IP addressing and internet routing.
3. Analyze the working of networked applications such as DNS, mail file transfer and www.
4. Implement client-server socket-based networked applications.
5. Implement the use of Computer Networks for Large scale Networked systems

UNIT-I

Data Communications: Components, analog and digital signals and Encoders, Modems, S232 Interfacing. Switching: Circuit Switching, Message Switching and Packet Switching. Topologies - Concept of layering - Protocols and Standards - ISO./ OSI model, TCP/IP.

UNIT -II

Data Link Layer: Error Control: Error detection and correction (CRC and Hamming code for single bit correction). Flow Control: stop and wait - - sliding window protocols-go Back-NARQ – selective repeat ARQ. MAC LAYER: Ethernet IEEE 802.3 LAN, Manchester encoding, Binary exponential algorithm, Efficiency calculation, ARP and RARP.

UNIT - III

Network Layer : Internetworks - virtual circuit and Datagram approach Routing - Distance Vector Routing ,Link State Routing , OSPF and BGP IPv4, addressing, Subnetting, IPv6, CIDR, ICMP and IGMP protocols.

UNIT - IV

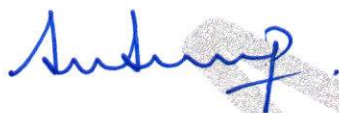
Transport Layer: Services of transport layer, Multiplexing and crash recovery. Transmission Control Protocol (TCP) - TCP window management Congestion Control, timer management and User Datagram Protocol (UDP). BE: COMPUTER SCIENCE AND ENGINEERING With effect from Academic year 2024-25

UNIT - V

Socket Programming: Primitive and advanced system calls, client/server iterative and concurrent programs IO multiplexing, Asynchronous IO and select system call. Application Layer: Domain Name Space (DNS)-SMTP -FTP-HTTP.

References:

1. Computer Networks, 5th Edition, Andrew S. Tanenbaum, David J. Wetherall, Pearson Education, 2021.
2. Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, Elsevier, 5th Edition, 2021.
3. Computer Networking: A Top-Down Approach, 6th Edition, James F. Kurose, Keith W. Ross, Pearson, 2022.




PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-LCC202**Database Systems Lab****Credits: 2**

Course Objectives: After completing this laboratory course, students will be able to:

1. Understand database design principles and develop Entity-Relationship (ER) models and relational schemas for real-world applications.
2. Apply SQL commands and integrity constraints for database creation, manipulation, and querying.
3. Design and implement advanced database objects such as procedures, functions, triggers, and indexes.
4. Analyse normalization, transaction management, concurrency control, and recovery mechanisms in database systems.
5. Explore NoSQL databases and compare relational and non-relational database models for modern data management applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Design ER diagrams, relational schemas, and database structures by applying database modeling concepts and integrity constraints.
2. Create and manage databases using SQL DDL and DML commands, joins, set operations, aggregate functions, and nested queries.
3. Develop stored procedures, functions, triggers, and indexes to enhance database functionality and performance.
4. Apply normalization techniques, transaction control operations, concurrency control methods, and recovery procedures to ensure database consistency and reliability.
5. Implement NoSQL database operations using MongoDB or Cassandra and compare their features, query mechanisms, and applications with relational Databases

Lab Experiments

1. Design an Entity-Relationship (E-R) diagram for a Banking Management System and convert it into relational schemas
2. Develop a database schema for a Library Management System by identifying entities, attributes, relationships, and constraints.
3. Create a database for a Student Information System and demonstrate the use of primary keys, foreign keys, and integrity constraints.
4. Create tables for a College Database using SQL Data Definition Language (DDL) commands and enforce key constraints.
5. Perform SQL operations using:
 - a. SELECT
 - b. INSERT
 - c. UPDATE
 - d. DELETE
 - e. WHERE
 - f. ORDER BY
 - g. GROUP BY

6. on a sample Employee database.
7. Write SQL queries using joins, set operations (UNION, INTERSECT, EXCEPT), aggregate functions, and nested subqueries.
8. Implement integrity constraints (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, NOT NULL) and test their enforcement.
9. Create SQL procedures, functions, and triggers for a Banking or Student Database application.
10. Perform normalization of a given relation up to 3NF/BCNF by identifying functional dependencies and decomposition steps.
11. Create and analyse different types of indexes (Single-column, Composite, and B+ Tree indexes) and compare query performance.
12. Implement transaction management using COMMIT, ROLLBACK, and SAVEPOINT commands on a Banking Database.
13. Simulate concurrent transactions and demonstrate serializability, recoverability, and isolation levels.
14. Implement lock-based concurrency control and demonstrate deadlock detection and resolution techniques.
15. Simulate log-based recovery techniques for transaction failures and analyse database recovery procedures.
16. Install and use a NoSQL database (MongoDB/Cassandra) to:
 - a) Create collections/tables
 - b) Insert and retrieve documents
 - c) Perform aggregation queries
 - d) Compare NoSQL operations with SQL databases.

Mini Project (Comprehensive DBMS Application)**Database-Driven Management System**

Develop a complete application such as:

- Student Management System
- Library Management System
- Hospital Management System
- Banking Management System

The project should include:

- E-R Diagram Design
- Relational Schema Design
- SQL Queries and Views
- Stored Procedures/Triggers
- Transactions and Concurrency Handling
- Indexing for Performance Optimization
- NoSQL Integration (Optional)
- Front-end Interface using Python/Java/Web Technologies

Suggested Tools: MySQL, PostgreSQL, Oracle, MongoDB, Python (Tkinter/Streamlit), PHP, Java, WT




P.V. Subbarao
PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



OM-LCC204**Machine Learning Lab****Credits: 2**

Course Objectives: After completing this laboratory course, students will be able to:

1. Apply data pre-processing, exploratory data analysis, and end-to-end machine learning workflows for solving real-world problems.
2. Develop and evaluate supervised learning models for regression and classification tasks using appropriate performance metrics.
3. Implement decision tree and ensemble learning techniques to improve predictive accuracy and model robustness.
4. Apply dimensionality reduction and unsupervised learning techniques for pattern discovery, clustering, and anomaly detection.
5. Design, train, evaluate, and deploy artificial neural network models using Scikit-Learn, PyTorch, and Streamlit.

Course Outcomes: Upon successful completion of the laboratory course, students will be able to:

1. Perform data pre-processing, feature engineering, exploratory data analysis, and build complete machine learning pipelines for real-world datasets.
2. Implement and evaluate regression and classification models including Linear Regression, Logistic Regression, binary classification, multiclass classification, and multilabel classification techniques.
3. Develop Decision Tree, Random Forest, Bagging, Boosting, and Voting Classifier models and compare their performance using suitable evaluation metrics.
4. Apply Principal Component Analysis (PCA), K-Means Clustering, and Gaussian Mixture Models (GMMs) for dimensionality reduction, clustering, and anomaly detection tasks.
5. Build, train, evaluate, and deploy Artificial Neural Networks and image classification models using Scikit-Learn, PyTorch, and Streamlit for end-to-end machine learning applications.

Lab Experiments

1. Perform Exploratory Data Analysis (EDA) on a real-world dataset using Pandas and Matplotlib to identify patterns, correlations, and missing values.
2. Implement data pre-processing techniques including handling missing values, feature scaling, encoding categorical variables, and train-test splitting.
3. Build an end-to-end machine learning pipeline that includes data collection, pre-processing, model training, prediction, and performance evaluation.

4. Train a Binary Classifier using the MNIST dataset to distinguish a specific digit and evaluate its performance using confusion matrix, precision, recall, and F1-score.
5. Implement Linear Regression using Scikit-Learn and compare the performance of Ordinary Least Squares and Regularized Regression (Ridge/Lasso).
6. Develop a Logistic Regression model for a classification problem and perform error analysis for multiclass and multilabel classification tasks.
7. Implement a Decision Tree classifier, visualize the generated tree, and analyze the effects of Gini Impurity and Entropy criteria.
8. Develop a Random Forest classifier and compare its performance with a single Decision Tree on a standard dataset.
9. Implement Ensemble Learning techniques including Bagging, Voting Classifiers, and Boosting, and compare their prediction accuracy.
10. Apply Principal Component Analysis (PCA) for dimensionality reduction and compare model performance before and after PCA.
11. Implement K-Means Clustering and determine the optimal number of clusters using the Elbow Method and Silhouette Score.
12. Develop a Gaussian Mixture Model (GMM) for clustering and anomaly detection and compare its performance with K-Means.
13. Build and train a Multi-Layer Perceptron (MLP) using Scikit-Learn for a classification problem and evaluate its performance.
14. Implement Linear Regression using PyTorch and compare the results with the Scikit-Learn implementation.
15. Develop an Image Classification model using PyTorch on the MNIST dataset using mini-batch gradient descent and evaluate the trained model.

Mini Project: End-to-End Machine Learning Application with Deployment

Develop a complete machine learning solution that includes:

- Data Collection and Pre-processing
- Exploratory Data Analysis
- Feature Engineering
- Model Selection and Training
- Hyperparameter Tuning
- Model Evaluation
- Deployment using Streamlit

Suggested Projects

1. House Price Prediction System
2. Student Performance Prediction
3. Loan Approval Prediction
4. Customer Churn Prediction
5. Disease Prediction System
6. Employee Attrition Prediction
7. Sales Forecasting System
8. Handwritten Digit Recognition System

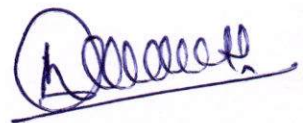
Tools and Libraries

- Python
- NumPy
- Pandas
- Matplotlib
- Scikit-Learn
- PyTorch
- Streamlit
- Jupyter Notebook / Google Colab



PROFESSOR

Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.

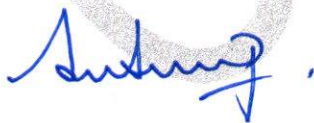


OM-SIP310**Summer Internship / Mini Project****Program Description**

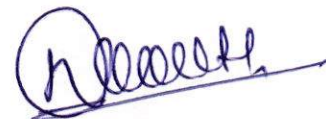
The Internship Program/ Mini Project allows MCA students to gain practical experience in the workplace before receiving their Post graduate degree. The internship is a required academic course. The student identifies companies willing to hire him/her on a full-time basis for 6-week period (minimum required) in the summer vacation. The evaluation committee evaluates the students and awards academic credits (1) in third semester. Those students who wish to do a Mini Project can use Problem statements and Data Sources from good quality sources and implement a solution. The student will be evaluated based on the working system that is presented in Semester III of this program.

Intended Learning Outcomes

1. Communicate a practical understanding of how a technology actually operates.
2. Demonstrate the ability to integrate and apply theoretical knowledge and skills developed in various courses to real-world situations in a business organization.
3. Exhibit the ability to effectively work in a professional environment and demonstrate work ethic and commitment in a work-based environment.
4. Demonstrate the ability to successfully complete internship assignments.
5. Reflect on personal and professional development needs and set strategic goals for advancing along an intended career path.
6. Communicate effectively in a professional environment in both English and regional language, orally and in writing.




PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University
Hyderabad-500 007.



Model Question Paper Format for CIE
MASTER OF COMPUTER APPLICATIONS (MCA)
Omega Institute of Science and Technology
M.C.A.-- Semester Examination, -----

Subject: -----

Time: 1 Hours

Max. Marks: 20

NOTE:

i) Answer all questions from Part - A, & any five questions from Part -B, choosing one question from each unit.

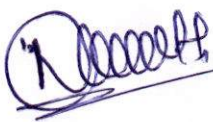
ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART -A (5Q x 2M =10 Mark)

S.NO		Question	Marks	BT	CO
1	a	Theory/ Problem			
	b	Theory/ Problem			
	c	Theory/ Problem			
	d	Theory/ Problem			
	e	Theory/ Problem			

PART-B (2Q x 5M= 10 Marls)

S.NO		Question	Marks	BT	CO
1	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
2	a	Theory/ Problem			
	b	Theory/ Problem			
3.	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
4	a	Theory/ Problem			
	b	Theory/ Problem			



P.V. Sushma

PROFESSOR

Department of Computer Science & Engineering
 University College of Engineering (A)
 Osmania University
 Hyderabad-500 007.




Model Question Paper Format for SEE
MASTER OF COMPUTER APPLICATIONS (MCA)

Omega Institute of Science and Technology

M.C.A .-- Semester Examination,-----

Subject:-----

Time: 1 Hours

Max. Marks: 70

NOTE: i) Answer all questions from Part - A, & any five questions from Part -B, choosing one question from each unit.

ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART -A (5Q x 2M =10 Mark)

S.NO		Question	Marks	BT	CO
1	a	UNIT-I			
	b	UNIT-I			
	c	UNIT-II			
	d	UNIT-II			
	e	UNIT-III			
	f	UNIT-III			
	g	UNIT-IV			
	h	UNIT-IV			
	i	UNIT-V			
	j	UNIT-V			

PART-B (5Q x 10M=50 Marls)

S.NO		Question	Marks	BT	CO
UNIT-I					
1	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
2	a	Theory/ Problem			
	b	Theory/ Problem			
UNIT-II					
3.	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
4	a	Theory/ Problem			
	b	Theory/ Problem			
UNIT-III					
5	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
6	a	Theory/ Problem			
	b	Theory/ Problem			
UNIT-IV					
7	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
8	a	Theory/ Problem			
	b	Theory/ Problem			
UNIT-V					
9	a	Theory/ Problem			
	b	Theory/ Problem			
(OR)					
10	a	Theory/ Problem			
	b	Theory/ Problem			

[Signature]

P.V. Sudha
PROFESSOR

Department of Computer Science & Engineering
 University College of Engineering (A)
 Osmania University
 Hyderabad-500 007.

[Signature]

[Signature]

OMISTIA