

Osmania University

Faculty of Informatics

Master of Computer Applications (MCA)

CBCS (Choice-Based Credit System) Syllabi

Semester I and II

With effect from the academic year 2026-2027

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



**Faculty of Informatics
Osmania University
2026-2027**

Contents

MCA Program Outcomes	3
Sustainable Development Goals(SDG)- Courses Mapping Matrix.....	5
Semester I.....	5
Semester II	6
General Course Structure.....	7
SCHEME OF INSTRUCTION	1
Semester – I	1
Semester – II	1
Detailed Syllabus for Semester II	23
Model Question Paper Format for CIE.....	49
Model Question Paper Format for SEE	50


Dean,
Faculty of Informatics
Osmania University

MCA Program Outcomes

PO1	Computational Knowledge: Apply knowledge of Computing fundamentals, Computing specialization, Mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and Conceptualization of computing models from defined problems and requirements.
PO2	Problem analysis: Identify, formulate, research literature, and solve complex Computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing sciences, and relevant domain disciplines.
PO3	Design /Development of Solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO5	Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PO6	Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.
PO7	Life-long Learning: Recognize the need, and have the ability, to engage in independent learning for continual development as a Computing professional.
PO8	Project management and finance: Demonstrate knowledge and understanding of computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO9	Communication Efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO10	Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the

P.V. Sudhakar

CBOS & Dean | Faculty of Informatics

Dean,
Faculty of Informatics
Osmania University

	consequential responsibilities relevant to professional computing practice.
PO11	Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
PO12	Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.


*Dean,
Faculty of Informatics
Osmania University*

Sustainable Development Goals(SDG)- Courses Mapping Matrix

Semester I

Course	SDG4	SDG8	SDG 9	SDG 11	SDG 12	SDG 13	Remarks
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

- 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.

P. V. Sudhakar

Dean,
Faculty of Informatics
Osmania University

Semester II

Course	SDG 4	SDG 8	SDG 9	SDG 11	SDG 12	SDG 13	Remarks
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

- 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

P. V. Gredha

Dean,
Faculty of Informatics
Osmania University

General Course Structure

Definition of Credit

Type	Credit
1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. 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
HPW	Hours Per Week
L	Lecture Hours/Week
T	Tutorial Hours/Week
P	Practical Hours/Week
Cr	Credits

Scheme of Examination

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

Credit Table R26(2026-2028)

Semester	Credits
I	24
II	24
III	28
IV	22
Total	98

P.V. Sridhar

Dean,
Faculty of Informatics
Osmania University

SCHEME OF INSTRUCTION

Master of Computer Applications

Semester – I

Course Code	Course Title	Credits	HPW (L/P)	Max Marks (CIE + SEE)
Theory				
PCC101	Mathematical Foundations of Computer Science	3	3L	30 + 70
PCC102	Data Structures using C	3	3L	30 + 70
PCC103	Object Oriented Programming Using Python	3	3L	30 + 70
PCC104	Operating Systems	3	3L	30 + 70
PCC105	Artificial Intelligence	3	3L	30 + 70
MGC106	Managerial Economics and Accountancy	3	3L	30 + 70
Practical				
LCC102	Data Structures using C Lab	2	4P	25 + 50
LCC103	Python Programming Lab	2	4P	25 + 50
HSC110	Soft Skills Lab	2	4P	25 + 50
Total		24	30	255+570

Semester – II

Course Code	Course Title	Credits	HPW (L/P)	Max Marks (CIE + SEE)
Theory				
PCC201	Applied Mathematics	3	3L	30 + 70
PCC202	Database Systems	3	3L	30 + 70
PCC203	Design and Analysis of Algorithms	3	3L	30 + 70
PCC204	Machine Learning	3	3L	30 + 70
PCC205	Fundamentals of Data Engineering	3	3L	30 + 70
PCC206	Computer Networks	3	3L	30 + 70
Practical				
LCC202	Database Systems Lab	2	4P	25 + 50
LCC203	Design and Analysis of Algorithms Lab	2	4P	25 + 50
LCC204	Machine Learning Lab	2	4P	25 + 50
SIP310	Summer Internship/ Mini Project *			
Total		24	30	255+570

*To be completed after the second semester, with credits to be awarded in the third semester.

P. V. Sudhakar

CBoS & Dean | Faculty of Informatics

*Dean,
Faculty of Informatics
Osmania University*

Detailed Syllabus for Semester I

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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

Course Contents

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.

P.V. Subbarao

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.

P.V. Sridhar

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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****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.


CBoS & Dean | Faculty of Informatics

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.



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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

P. V. Sridhar

CBoS & Dean | Faculty of Informatics

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.



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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****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. Sudhakar

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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****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.



CBoS & Dean | Faculty of Informatics

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, Tata McGraw-Hill, 2008.
3. 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

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

P.V. Giddula

CBoS & Dean | Faculty of Informatics

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 understand types of demand and factors related to it.
2. Identify different types of markets and determine price-output under perfect competition.
3. Determine working capital requirements and payback.
4. Analyze and interpret financial statements through ratios.
5. Determine double entry bookkeeping methods

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****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.

P.V. Sudha
CBoS & Dean | Faculty of Informatics

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 Educational Publishers, 2011.
2. Maheswari S.N., Introduction to Accountancy, Vikas Publishing House, 2005.
3. Pandey I.M., Financial Management, Vikas Publishing House, 2009.
4. S.P. Jain and K.L. Narang, Financial Accounting, Kalyan Publishers, 2018.
5. M. Hanif and A. Mukherjee, Modern Accountancy, McGraw Hill, 3rd Edition, 2018.



CBoS & Dean | Faculty of Informatics

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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping


CBoS & Dean | Faculty of Informatics

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:
 - Singly Linked List
 - Circular Linked List
 - Doubly Linked Listincluding insertion, deletion, and traversal operations.
10. Implement Binary Tree creation and perform Preorder, Inorder, and Postorder traversals.
11. Develop a program to create a Binary Search Tree (BST) and perform insertion, deletion, and searching operations.
12. Implement Graph representation using adjacency matrix/adjacency list and perform Breadth First Search (BFS) and Depth First Search (DFS) traversals.
13. Implement Linear Search and Binary Search algorithms and compare their performance for different input sizes.

P. V. Gidder

CBoS & Dean | Faculty of Informatics

14. Develop programs for the following sorting techniques and compare their execution times:

- Selection Sort
- Bubble Sort
- Insertion Sort
- Merge Sort
- Quick Sort

15. Implement a Hash Table with collision resolution techniques such as:

- Linear Probing
- Quadratic Probing
- Chaining

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).



CBoS & Dean | Faculty of Informatics

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.

CO-PO Mapping Matrix

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

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

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

P. V. Sudha

CBoS & Dean | Faculty of Informatics

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.


CBoS & Dean | Faculty of Informatics

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.