

**Scheme and Syllabus of
Master of Computer Application
(02-Year Post Graduate Program)**

(2025-2026 onwards)

By

**Skill Department of Computer Science/ IT
Under
Skill Faculty of Engineering and Technology**



**Shri Vishwakarma Skill University
Dudhola, Palwal-121102, Haryana**

Monika
Dr. Monika Jangra
(Skill Assistant Professor)
Skill Department of CS/IT

Developed By:

Madhuri
Ms. Madhuri Saini
(Skill Instructor)
Skill Department of CS/IT

Verified By:

Usha
Prof. (Dr) Usha Batra
Chairperson
Skill Department of CS/IT

Aashish
Prof. (Dr) Aashish Shrivastava
Dean

Skill Faculty of Engineering and Technology,
Shri Vishwakarma Skill University, Palwal

Prof. (Dr) Suresh Kumar
Dean Academics
Shri Vishwakarma Skill University, Palwal

1.1 VISION and MISSION

Vision of the University

To emerge as one of the foremost institutions of quality in skill education acknowledges by industry, nationally and internationally.

Mission of the University

To focus on quality of skill education, aspired by students and admired by industry, nationally and internationally.

Vision of the Department

To build highly skilled and competent professionals with a futuristic outlook for research and innovation who can also generate sustainable employment with virtuous professional values.

Mission of the Department

1. To impart skill education and training as per industry needs in the domain of Computer Science & Engineering.
2. To develop and maintain the state of art infrastructure for building sustainable skilled IT ecosystem in the domain.
3. To cultivate the environment for fostering new ideas, research and innovations leading to emergence of skilled entrepreneurs, leaders, innovators and professionals.

1.2 PROGRAM OUTCOME

This program is aimed at skilling the students for the jobs in IT industry at diploma level and aims at building the following key competencies amongst the learner.

PO1: Computational Knowledge

Apply knowledge of computing fundamentals, computing specialisation, mathematics, and domain knowledge appropriate for the computing specialisation to the abstraction and conceptualisation 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 the 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 practices.

PO7: Life-long Learning

Recognise 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 the 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 consequential responsibilities relevant to professional computing practices.

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 use innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

1.3 PROGRAM SPECIFIC OUTCOMES

PSO1: Graduates will be able to apply analytical and logical reasoning, along with programming knowledge, to design and develop efficient software solutions for real-world problems.

PSO2: Graduates will demonstrate the ability to apply theoretical and practical knowledge of computer applications to build innovative software products and explore entrepreneurial ventures.

PSO3: Graduates will possess effective communication skills, enabling them to convey technical and non-technical information clearly through oral, written, and digital formats.

PSO4: Graduates will be capable of working effectively in structured environments, demonstrating responsibility and ethical behavior while collaborating within defined roles and boundaries.

Navika
medhavi

1.4 PROGRAM EDUCATIONAL OBJECTIVES

PEO1: Graduates will be equipped with a strong foundation in computer science and applications, combined with professional and entrepreneurial skills, to enable successful careers in the IT industry, pursue higher education, or establish their own ventures.

PEO2: Graduates will demonstrate the ability to work collaboratively in multidisciplinary teams, contribute to software design and implementation, communicate effectively in professional settings, and assist in project documentation and reporting.

PEO3: Graduates will recognize the importance of ethical responsibilities, environmental sustainability, and societal impact, and practice their profession with integrity and awareness of global challenges.

The PO and PSO mapping, PO and PEO mapping are given in table 3 and table 4.

The mapping is defined such as 0: no mapping; 1: Low mapping; 2: Average mapping; 3: Complete mapping.

Table 3. PO and PSO mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PSO1	2	0	3	0	1	0	0	0	0	0	0	0
PSO2	1	2	0	0	3	0	0	0	0	0	3	0
PSO3	0	0	0	2	0	0	0	0	1	3	0	0
PSO4	0	0	0	0	0	2	1	1	3	0	0	2

Table 4. PO and PEO mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PEO1	3	1	3	1	2	0	0	0	0	0	0	0
PEO2	0	0	0	0	0	2	0	3	3	3	0	1
PEO3	0	2	0	0	0	1	1	0	0	0	0	0

*Monika
modhwal*

1.5 DOMAIN WISE CREDIT BREAKUP

Sr. No.	Category	SVSU MCA Program Credits
1	Program Core Courses (PC)	58
2	¹ Ability Enhancement Course (AEC) [courses related to communication and professional ethics, aptitude, leadership, wellness, innovation, digital skills, civic awareness, and sustainable thinking, etc.]	04
3	² Value Added Course (VAC)	06
4	Program Elective (PE)	09
5	Project work, seminar and internship in industry or elsewhere	07
6	³ Audit Courses (AU) [courses related to Environmental Sciences, Induction training, Indian Constitution, Essence of Indian Traditional Knowledge, Yoga and Health etc.]	(non-credit)
	Total	94
<i>Note: PC includes PE credits also</i> <i>^{1,2,3}Ability Enhancement Courses and Audit Courses may be decided by the time offering. The credits may vary slightly, given that minimum graduation credit criteria fulfil.</i> <i>Bridge courses may be offered from other programs as appropriate.</i>		

1.6 SEMESTER WISE SCHEME

The semester wise scheme is given as

SEMESTER I

Subject code	Subject Title	hours per week			Credit	Category
		L	T	P		
24PENG01	Professional Communication	1	0	0	1	AEC
24PENG02	Professional Communication Practical	0	0	2	1	AEC
25PMCA01	Mathematical Foundations of Computer Applications	3	1	0	4	PC
25PMCA02	Data Mining and Applications	3	0	0	3	PC
25PMCA03	Data Mining and Applications Lab	0	0	2	1	PC
25PMCA04	Programming with C	3	0	0	3	PC
25PMCA05	Programming with C Lab	0	0	2	1	PC
25PMCA06	Fundamentals of Data Science	3	1	0	4	PC
24UYHS01	Yoga and Health	2	0	0	0	AU
		15	2	6	18	
Total hours		23				

SEMESTER II

Subject code	Subject Title	hours per week				Category
		L	T	P	Credit	
25PMCA07	Advanced Data Structures	3	1	0	4	PC
25PMCA08	Advanced Data Structures Lab	0	0	2	1	PC
25PMCA09	Object Oriented Programming using Python	3	0	0	3	PC
25PMCA10	Object Oriented Programming using Python Lab	0	0	2	1	PC
25PMCA11	Distributed Systems	3	1	0	4	PC
25PMCA12	Cyber Security and Information Assurance	3	0	0	3	PC
	Value Added Course-I	2	0	0	2	VAC
		14	2	4	18	
Total hours		20				
After Sem II, student will undergo internship that will be evaluated in Sem III.						

SEMESTER III

Subject code	Subject Title	hours per week			Credit	Category
		L	T	P		
25PMCA13	Program Elective I	3	1	0	4	PC
25PMCA14	Program Elective I Lab	0	0	2	1	PC
25PMCA15	Program Elective II	3	0	0	3	PC
25PMCA16	Program Elective II Lab	0	0	2	1	PC
25PMCA17	Software Testing and Quality Assurance	3	0	0	3	PC
25PMCA18	Excel & Power BI for Data Visualization	0	0	2	1	VAC
25PMCA19	Capstone Project	0	0	10	5	Project
	Audit course-I	2	0	0	0	AU
25PMCA20	Internship (credits from sem II)	0	0	0	2	Project
		11	1	16	20	
	Total hours	28				

*Navika
medhvi*

SEMESTER IV

Subject code	Subject Title	hours per week			Cred it	Catego ry
		L	T	P		
25PMCA18	Seminar	0	2	0	2	PC
	⁴ Value Added Course-II	2	0	0	2	VAC
25PMCA19	Major Project/Dissertation	0	0	2	12	PC
	Ability Enhancement Course	2	0	0	2	AEC
	Audit course-II	2	0	0	0	AU
		4	2	2	18	
Total hours		30				
⁴ Value added courses to be selected from other domain, may be decided at the time offering as appropriate.						

1.7 LIST OF ELECTIVES*

Program Elective I	Program Elective II
Cryptography and Network Security	Risk Analysis and Management
Artificial Intelligence	Intrusion Detection Systems
Information Retrieval	Generative and Explainable AI
Machine learning and Pattern Recognition	Explainable AI
Data Analytics	Adversarial AI

*More elective courses may also be added to the list of open electives and elective (Domain Specific) at the time of offering of courses by the Department.

Navika
modhvi

MCA Semester I Detailed Syllabus

Subject: Mathematical Foundations of Computer Applications (25PMCA01)

Category: Program Core | L-T-P: 3-1-0 | Credits: 4

Course Learning Outcomes (CLOs):

At the end the course, student will be able to

1. Understand fundamental mathematical concepts used in computing.
2. Apply set theory and logic to solve computational problems.
3. Analyze relations, functions, and combinatorics.
4. Use graph theory for modeling and solving problems.
5. Solve recurrence relations and understand their role in algorithms.

Unit-wise Syllabus:

Unit 1: Sets, Relations and Functions

Sets, finite and infinite sets, power set, universal set, set operations (union, intersection, difference, complement), Venn diagrams, laws of set theory, Cartesian product of sets, relations, types of relations (reflexive, symmetric, transitive, antisymmetric, equivalence), matrix and digraph representation of relations, closures of relations, partial ordering, Hasse diagrams, functions, types of functions (one-one, onto, bijective), composition of functions, inverse functions, identity function, binary operations

Unit 2: Propositional and Predicate Logic

Propositions, logical connectives, truth tables, tautology, contradiction, contingency, algebra of propositions, logical equivalence, laws of logic, conditional and bi-conditional statements, rules of inference, arguments, normal forms (CNF and DNF), predicates, quantifiers (universal and existential), free and bound variables, translating statements into predicate logic, rules of inference for quantified statements, direct proof, indirect proof, proof by contradiction, contrapositive

Unit 3: Combinatorics and Recurrence Relations

Addition principle, multiplication principle, permutations, combinations, permutations with repetition, combinations with repetition, binomial theorem, Pascal's triangle, pigeonhole principle, generalized pigeonhole principle, recurrence relations, linear recurrence relations with constant coefficients, homogeneous recurrence relations, non-homogeneous recurrence relations, generating functions

Unit 4: Graph Theory – Trees, Connectivity, Traversal

Graphs, types of graphs (simple, multigraph, directed, weighted), degree of vertex, path, cycle, connected graphs, disconnected graphs, adjacency matrix, incidence matrix, cut vertex, bridge, Eulerian path and circuit, Hamiltonian path and circuit, trees, rooted trees, binary trees, tree traversal (inorder, preorder, postorder), spanning trees, minimum spanning trees, Kruskal's algorithm, Prim's algorithm.

Navika
modhum

Unit 5: Algebraic Structures – Groups, Rings, Monoids, Lattices

Binary operations, closure, associativity, identity, invertibility, semigroups, monoids, submonoids, group definition, subgroups, cyclic groups, cosets, Lagrange's theorem, group homomorphism, group isomorphism, rings, commutative rings, rings with unity, subrings, ring homomorphism, integral domains, fields, posets, lattices, bounded lattices, complemented lattices, distributive lattices, modular lattices

Textbooks:

1. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", McGraw Hill
2. Tremblay and Manohar, "Discrete Mathematical Structures", McGraw Hill

Reference Books:

1. C.L. Liu, "Elements of Discrete Mathematics"
2. S. Lipschutz, "Discrete Mathematics", Schaum Series
3. J.P. Tremblay, "Mathematical Structure for Computer Science"

manika
manohar

Subject: Data Mining and Applications (25PMCA02)

Category: PC | L-T-P: 3-0-0 | Credits: 3

Course Learning Outcomes (CLOs):

At the end the course, student will be able to

1. Understand basic concepts of data mining and knowledge discovery.
2. Apply association rule mining, clustering and classification techniques.
3. Evaluate the performance of data mining models.
4. Use data pre-processing techniques for cleaning and transformation.
5. Implement basic mining algorithms on real-world datasets.

Unit-wise Syllabus:

Unit 1: Introduction to Data Mining

Data mining definition, knowledge discovery in databases (KDD) process, steps in KDD (selection, preprocessing, transformation, data mining, interpretation/evaluation), difference between KDD and data mining, types of data mining tasks (descriptive and predictive), data mining functionalities, challenges in data mining, major issues in data mining, applications of data mining in business, healthcare, finance, cybersecurity, and social media

Unit 2: Data Preprocessing

Data quality issues, need for data preprocessing, data cleaning techniques (handling missing values, noisy data, outliers), data integration (schema integration, entity identification, redundancy), data transformation (normalization, smoothing, aggregation, generalization), data reduction (dimensionality reduction, data cube aggregation, numerosity reduction), feature selection, discretization and concept hierarchy generation

Unit 3: Classification Techniques

Introduction to classification, supervised learning, decision tree induction, ID3 algorithm, attribute selection measures (information gain, gain ratio, Gini index), pruning techniques, rule-based classification, naïve Bayes classifier, Bayes theorem, advantages and limitations of decision trees and naïve Bayes, model evaluation metrics (accuracy, precision, recall, F1-score, confusion matrix), cross-validation

Unit 4: Clustering

Introduction to clustering, types of clustering (partitional, hierarchical, density-based), K-Means clustering, initialization and convergence, hierarchical clustering (agglomerative and divisive), distance measures (Euclidean, Manhattan, cosine), dendrograms, DBSCAN algorithm, density-reachability and density-connectivity, comparison of clustering algorithms, applications of clustering in pattern recognition, image analysis, customer segmentation

Unit 5: Association Rule Mining

Introduction to association rules, frequent itemsets, support and confidence, Apriori algorithm, candidate generation and pruning, limitations of Apriori, FP-Growth algorithm, pattern growth

approach, comparison of Apriori and FP-Growth, interestingness measures (lift, conviction, leverage), evaluation of association rules, applications of association rule mining in market basket analysis, recommendation systems

Textbooks:

1. Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques"
2. Pang-Ning Tan, "Introduction to Data Mining"

Reference Books:

1. Arun K. Pujari, "Data Mining Techniques"
2. Margaret Dunham, "Data Mining: Introductory and Advanced Topics"
3. Ian H. Witten, "Data Mining: Practical Machine Learning Tools and Techniques"

Subject: Data Mining and Applications Lab (25PMCA03)

Category: PC | L-T-P: 0-0-2 | Credits: 1

Course Learning Outcomes (CLOs):

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

1. Pre-process data for mining tasks.
2. Implement classification algorithms like ID3 and Naive Bayes.
3. Perform clustering using K-Means and DBSCAN.
4. Discover frequent patterns and generate association rules.
5. Use tools like WEKA/Python for implementing mining techniques.

List of Experiments:

- 1 Implement a program to demonstrate the KDD process using a sample dataset
- 2 Case study/report on data mining applications in domains like healthcare, retail, or social media
- 3 Handle missing values in a dataset using techniques like mean/mode imputation and deletion
- 4 Perform data cleaning and noise removal using outlier detection techniques
- 5 Implement data normalization techniques (Min-Max, Z-score)
- 6 Demonstrate dimensionality reduction using Principal Component Analysis (PCA)
- 7 Implement a Decision Tree classifier and visualize the tree structure
- 8 Build and evaluate a Naïve Bayes classifier using a real-world dataset
- 9 Compare performance of Decision Tree and Naïve Bayes using metrics and cross-validation
- 10 Implement K-Means clustering and visualize the clusters
- 11 Perform Hierarchical Clustering and generate a dendrogram
- 12 Apply DBSCAN clustering and interpret clusters based on density
- 13 Apply Apriori algorithm to a transactional dataset and generate frequent itemsets and rules
- 14 Implement FP-Growth algorithm and compare with Apriori in terms of performance
- 15 Evaluate association rules using support, confidence, lift, and conviction

Mouib
Mouib

Textbooks:

3. Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques"
4. Pang-Ning Tan, "Introduction to Data Mining"

Reference Books:

4. Arun K. Pujari, "Data Mining Techniques"
5. Margaret Dunham, "Data Mining: Introductory and Advanced Topics"
6. Ian H. Witten, "Data Mining: Practical Machine Learning Tools and Techniques"

Monika
Madhus

Subject: Programming with C (25PMCA04)

Category: PC | L-T-P: 3-0-0 | Credits: 3

Course Learning Outcomes (CLOs):

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

1. Understand structure and syntax of C programming language.
2. Develop programs using control structures and functions.
3. Apply arrays, strings, pointers and structures in solving problems.
4. Implement file handling in C.
5. Debug and test C programs effectively.

Unit-wise Syllabus:

Unit 1: Introduction to C

History and structure of C program, variables, constants, keywords, data types, type conversion and type casting, arithmetic operators, relational operators, logical operators, bitwise operators, increment and decrement operators, assignment operators, conditional operator, input and output functions (`scanf`, `printf`), basic program structure, compilation and execution.

Unit 2: Control Statements and Functions

Conditional statements (`if`, `if-else`, nested `if`, `switch`), loop control structures (`for`, `while`, `do-while`), `break`, `continue`, `goto`, function definition and declaration, function calling and parameter passing (call by value, call by reference), return values, recursion, scope and lifetime of variables, storage classes

Unit 3: Arrays, Strings, and Pointers

One-dimensional and two-dimensional arrays, array initialization and memory layout, strings and string handling functions (`strlen`, `strcpy`, `strcmp`, `strcat`), character arrays vs string literals, introduction to pointers, pointer arithmetic, pointers and arrays, pointers to strings, pointers to functions, pointers to arrays

Unit 4: Structures, Unions, and Dynamic Memory Allocation

Structure definition and declaration, accessing structure members, nested structures, array of structures, pointer to structure, structure as function argument, union definition and usage, differences between structures and unions, dynamic memory allocation functions (`malloc`, `calloc`, `realloc`, `free`), memory leaks and best practices

Unit 5: File Handling

Introduction to file handling, file streams, file opening and closing, reading from and writing to text files, file modes (`r`, `w`, `a`, `r+`, `w+`), formatted and unformatted I/O (`fscanf`, `fprintf`, `fgetc`, `fputc`, `fgets`, `fputs`), file error handling (`feof`, `ferror`), command-line arguments, practical examples using file handling.

Monika
14 | Page
anish

Textbooks:

1. E. Balagurusamy, "Programming in ANSI C"
2. Yashavant Kanetkar, "Let Us C"

Reference Books:

1. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language"
2. Byron Gottfried, "Programming with C", Schaum's Outline Series
3. Herbert Schildt, "C: The Complete Reference"

Subject: Programming with C Lab (25PMCA05)**Category: PC | L-T-P: 0-0-2 | Credits: 1****Course Learning Outcomes (CLOs):****At the end of the course, student will be able to**

1. Develop basic programs using input/output and operators.
2. Implement conditional and looping constructs.
3. Work with arrays, strings, and pointers in C.
4. Use structures and unions in programs.
5. Implement file handling operations.

List of practical:

- 1 Write a program to demonstrate the use of variables, data types, and arithmetic operators
- 2 Implement a program to take user input and display output using scanf and printf
- 3 Write a program using conditional statements (if, if-else, switch)
- 4 Write programs using different loops (for, while, do-while)
- 5 Implement a program to demonstrate the use of break, continue, and goto
- 6 Write a program using functions with parameters and return values
- 7 Implement recursion with examples like factorial, Fibonacci series
- 8 Write a program to declare and initialize arrays, and perform operations on them
- 9 Write a program to perform string operations using standard library functions (strlen, strcpy, etc.)
- 10 Implement a program demonstrating basic pointer operations and pointer arithmetic
- 11 Write a program to show the relationship between arrays and pointers
- 12 Write a program to define a structure and access its members
- 13 Write a program to implement dynamic memory allocation using malloc and free
- 14 Write a program to read from and write to a text file using fopen, fprintf, fscanf, etc.
- 15 Implement a program to accept and use command-line arguments

*Manika
Medhvi*

Subject: Fundamentals of Data Science (25PMCA06)

Category: PC | L-T-P: 3-1-0 | Credits: 4

Course Learning Outcomes (CLOs):

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

1. Understand the data science process and lifecycle.
2. Perform data wrangling and pre-processing.
3. Apply statistical techniques for data analysis.
4. Visualize and interpret data using basic tools.
5. Understand basics of machine learning techniques.

Unit-wise Syllabus:

Unit 1: Introduction to Data Science

Definition of Data Science, data types (structured, unstructured, semi-structured), levels of data measurement (nominal, ordinal, interval, ratio), components of data science, data science workflow (data collection, cleaning, exploration, modeling, interpretation), role of a data scientist, tools for data science (Python, R, Jupyter, etc.)

Unit 2: Data Wrangling

Handling missing data (deletion, imputation), data cleaning techniques, detecting and handling duplicates, data transformation (log transform, binning, encoding categorical variables), feature scaling techniques (normalization, standardization, min-max scaling, robust scaling), data type conversions, parsing dates and times, data integration

Unit 3: Exploratory Data Analysis (EDA)

Descriptive statistics (mean, median, mode, variance, standard deviation), data distributions (normal, skewed), data summarization using Pandas, frequency tables, cross-tabulation, correlation analysis, outlier detection, handling outliers, detecting patterns and anomalies, use of EDA in hypothesis generation

Unit 4: Data Visualization

Importance of data visualization, plotting with Matplotlib (line plot, bar chart, histogram, pie chart), Seaborn plots (heatmap, boxplot, scatter plot, violin plot), customizing plots (labels, titles, legends, themes), combining multiple plots, interactive plots with Plotly or Dash (optional), creating basic data dashboards

Unit 5: Introduction to Machine Learning

Definition of machine learning, difference between AI, ML, and data science, types of machine learning (supervised, unsupervised, reinforcement), overview of supervised learning (classification and regression), unsupervised learning (clustering), training and testing datasets, overfitting and underfitting, introduction to scikit-learn, simple ML workflow

Textbooks:

1. Joel Grus, "Data Science from Scratch"
2. Cathy O'Neil and Rachel Schutt, "Doing Data Science"

Reference Books:

1. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn"
2. Wes McKinney, "Python for Data Analysis"
3. Roger D. Peng, "Exploratory Data Analysis with R"

*Alcanta
medvni*

MCA Semester II Detailed Syllabus

Subject: Advanced Data Structures (24PMCA07)

Category: PC | L-T-P: 3-1-0 | Credits: 4

Course Learning Outcomes

After successful completion of the course, students will be able to:

1. Understand and apply advanced data structures in problem-solving.
2. Analyze the efficiency of different data structures.
3. Implement trees, heaps, and hash tables.
4. Solve problems using graphs and related algorithms.
5. Use appropriate data structures for real-world applications.

Unit-wise Syllabus:

Unit 1: Advanced Foundations of Data Structures

Abstract Data Types (ADTs), Amortized analysis, Average vs worst-case complexity, Recurrence relations, Master's theorem, Sparse arrays, Dynamic arrays, Multidimensional arrays, Skip lists, Unrolled linked lists, XOR linked lists, Memory-efficient linked structures, Applications in memory management and real-time systems

Unit 2: Advanced Stacks and Queues

Min-max stacks, K-stacks in an array, Parsing applications, Backtracking, Virtual machines, Deques, Input/output-restricted deques, Double-ended priority queues, Multilevel queues, Circular buffer implementation, Lock-free queues, Expression parsing and evaluation, Recursion optimization, Job scheduling, Memory buffering

Unit 3: Trees and Advanced Tree Structures

Binary trees, Morris traversal, Threaded traversal, AVL trees, Red-Black trees, Splay trees, Treaps, B-Trees, B+ Trees, Segment Trees, Fenwick Trees (Binary Indexed Trees), Tries, Patricia Trees, Applications in databases, Applications in indexing, Applications in file systems

Unit 4: Heaps, Hashing, and Priority Structures

Binomial Heaps, Fibonacci Heaps, Pairing Heaps, Soft Heaps, Mergeable heaps, Lazy deletion, Perfect hashing, Cuckoo hashing, Extendible hashing, Dynamic rehashing, Bloom filters, Applications in compilers, Applications in cache management, Applications in distributed systems

*Monika
Mishra*

Unit 5: Graph Algorithms and Advanced Applications

Adjacency maps, Compressed sparse row/column representations, BFS with optimizations, DFS with optimizations, Topological Sorting, Strongly Connected Components (Kosaraju, Tarjan), Dijkstra's algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm, Johnson's algorithm, A* search, Kruskal's algorithm, Prim's algorithm, Borůvka's algorithm, Maximum Flow (Ford-Fulkerson, Edmonds-Karp), Matching in Bipartite Graphs, Network Reliability, Applications in routing, Applications in social network analysis, Applications in recommendation systems

Textbooks

1. Ellis Horowitz & Sartaj Sahni, *Fundamentals of Data Structures in C*, Galgotia.
2. Narasimha Karumanchi, *Data Structures and Algorithms Made Easy*, CareerMonk.

Recommended Books

1. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, Pearson.
2. Seymour Lipschutz, *Data Structures (Schaum's Outline)*, McGraw Hill.
3. Yedidyah Langsam et al., *Data Structures Using C and C++*, PHL.

Subject: Advanced Data Structures Lab (24PMCA08)

Category: PC | L-T-P: 0-0-2 | Credits: 1

Course Learning Outcomes (CLOs):

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

1. Develop basic programs using input/output and operators.
2. Implement conditional and looping constructs.
3. Work with arrays, strings, and pointers in C.
4. Use structures and unions in programs.
5. Implement file handling operations.

List of Practical:

1. Implement insertion, deletion, searching, and traversal in a static array.
2. Write a program to count the number of operations for linear and binary search to analyze time complexity.
3. Implement creation, insertion, deletion, and traversal operations.
4. Implement forward and backward traversal with insertion and deletion at both ends.
5. Implement circular linked list with operations – insert at end, delete from beginning, and traversal.
6. Implement push, pop, and peek operations.
7. Implement stack operations using dynamic memory allocation.
8. Convert an infix expression to postfix and evaluate it using stack.
9. Implement factorial and Fibonacci series using recursion and compare with iterative solutions.

Handwritten signature:
Munida
Munida

10. Implement linear queue, circular queue, and priority queue using arrays.
11. Construct a binary tree and perform insertion and deletion.
12. Implement in-order, pre-order, and post-order traversals (both recursive and non-recursive).
13. Construct a BST and implement search, insertion, and deletion operations.
14. Implement rotation operations (LL, RR, LR, RL) in AVL tree balancing.
15. Implement creation and in-order traversal of a threaded binary tree.
16. Implement max-heap and min-heap creation.
17. Implement heap sort using max-heap.
18. Implement priority queue using heap data structure.
19. Implement hash table with linear probing for collision resolution.
20. Implement hash table using chaining technique and demonstrate rehashing when load factor increases.

Textbooks

1. Ellis Horowitz & Sartaj Sahni, *Fundamentals of Data Structures in C*, Galgotia.
2. Narasimha Karumanchi, *Data Structures and Algorithms Made Easy*, CareerMonk.

Recommended Books

1. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, Pearson.
2. Seymour Lipschutz, *Data Structures (Schaum's Outline)*, McGraw Hill.
3. Yedidyah Langsam et al., *Data Structures Using C and C++*, PHI.

Subject: Object Oriented Programming using Python (25PMCA09)

Category: PC | L-T-P: 3-0-0 | Credits: 3

Course Learning Outcomes

After successful completion of the course, students will be able to:

1. Understand the object-oriented approach in Python.
2. Create and use classes and objects efficiently.
3. Apply encapsulation, inheritance, and polymorphism.
4. Work with files and handle exceptions.
5. Build modular, maintainable Python applications.

Unit-wise Syllabus:

Unit 1: Introduction to Python and OOP Concepts

Python basics, variables, data types, operators, control structures, features of object-oriented programming.

Naika
20 | P a g e
medhvi

Unit 2: Functions and Modules

Function definition and calling, recursion, default and keyword arguments, modules and packages, importing modules.

Unit 3: Classes and Objects

Creating classes and objects, `__init__` method, instance vs class variables, `self` keyword, access specifiers.

Unit 4: Inheritance and Polymorphism

Types of inheritance, method overriding, `super()` function, abstract classes, duck typing, operator overloading.

Unit 5: File Handling and Exception Management

Opening and reading/writing files, file modes, exception types, `try-except-finally` blocks, raising exceptions, `with` statement.

Textbooks

1. Timothy A. Budd, *Exploring Python*, McGraw Hill.
2. E. Balagurusamy, *Introduction to Computing and Problem Solving with Python*, McGraw Hill.

Recommended Books

1. Mark Lutz, *Learning Python*, O'Reilly.
2. R. Nageswara Rao, *Core Python Programming*, Dreamtech Press.
3. Paul Barry, *Head First Python*, O'Reilly.

Subject: Object Oriented Programming using Python Lab (25PMCA10)

Category: PC | L-T-P: 0-0-2 | Credits: 1

Course Learning Outcomes (CLOs):

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

1. Develop Python programs using basic programming constructs and control structures.
2. Apply modular and reusable coding practices using functions and modules.
3. Implement object-oriented programming concepts such as classes, inheritance, and polymorphism in Python.
4. Use file handling and exception management techniques for robust Python applications.
5. Design and build Python-based solutions applying real-world programming principles.

List of Practical Experiments

1. Write a Python program to demonstrate basic I/O operations, use of variables, and different data types.
2. Implement a program using arithmetic, logical, and comparison operators with control flow statements (`if`, `elif`, `else`).

*Maarika
medhu*

3. Write a Python program using `for` and `while` loops with `break`, `continue`, and `pass` statements.
4. Define and call user-defined functions with default, positional, and keyword arguments.
5. Implement a recursive function to compute factorial or Fibonacci numbers.
6. Create and use Python modules and packages, including importing built-in and user-defined modules.
7. Write a class with an `__init__()` method and demonstrate the use of `self` and instance variables.
8. Differentiate between instance and class variables using an example with multiple objects.
9. Demonstrate access specifiers (`public`, `_protected`, `__private`) through a Python class.
10. Implement single and multiple inheritance, and override a method in the child class.
11. Use the `super()` function to invoke parent class constructors and methods.
12. Create an abstract class using `abc` module and implement its abstract methods in a subclass.
13. Demonstrate duck typing and operator overloading using custom Python classes.
14. Write a program for file handling using `open()`, `read()`, `write()`, and different file modes.
15. Implement exception handling using `try-except-finally`, `raise`, and `with` state

Textbooks

3. Timothy A. Budd, *Exploring Python*, McGraw Hill.
4. E. Balagurusamy, *Introduction to Computing and Problem Solving with Python*, McGraw Hill.

Recommended Books

4. Mark Lutz, *Learning Python*, O'Reilly.
5. R. Nageswara Rao, *Core Python Programming*, Dreamtech Press.
6. Paul Barry, *Head First Python*, O'Reilly.

Subject: Distributed Systems (25PMCA11)

Category: PC | L-T-P: 3-1-0 | Credits: 4

Course Learning Outcomes

After successful completion of the course, students will be able to:

1. Understand principles and models of distributed computing.
2. Apply communication and synchronization techniques.
3. Evaluate naming, consistency, and replication strategies.
4. Analyze distributed file systems and fault tolerance.
5. Explore modern distributed architectures.

Unit-wise Syllabus:

Unit 1: Fundamentals of Distributed Systems

Definition, goals, types of distributed systems, client-server and peer-to-peer models, transparency, scalability.

Unit 2: Communication in Distributed Systems

Message passing, remote procedure call (RPC), remote method invocation (RMI), group communication, multicast.

Unit 3: Synchronization and Coordination

Logical clocks, Lamport timestamps, vector clocks, mutual exclusion algorithms, election algorithms, distributed transactions.

Unit 4: Naming and File Systems

Name services, distributed file system architecture, file access, file sharing, replication, consistency models.

Unit 5: Fault Tolerance and Emerging Trends

Failure models, reliable communication, recovery, replication strategies, blockchain basics, cloud and edge computing overview.

Textbooks

1. Andrew S. Tanenbaum, *Distributed Systems: Principles and Paradigms*, Pearson.
2. George Coulouris et al., *Distributed Systems: Concepts and Design*, Pearson.

Recommended Books

1. M.L. Liu, *Distributed Computing: Principles and Applications*, Pearson.
2. Pradeep K. Sinha, *Distributed Operating Systems*, PHI.
3. Rajiv Chopra, *Distributed Systems*, S. Chand.

Subject: Cyber Security and Information Assurance (25PMCA12)

Category: PC | L-T-P: 3-0-0 | Credits: 3

Course Learning Outcomes

After successful completion of the course, students will be able to:

1. Identify various types of cyber threats and attacks.
2. Understand network security and secure communication.
3. Apply cryptographic techniques.
4. Implement access control and authentication methods.
5. Analyze legal and ethical aspects of cybersecurity.

Unit-wise Syllabus:**Unit 1: Introduction to Cyber Security**

Cyber security fundamentals, cyber crimes, types of attacks – phishing, malware, DoS, advanced persistent threats.

*Review
medhu*

Unit 2: Network Security

Basics of networking, firewalls, IDS and IPS, secure protocols, wireless security, VPNs.

Unit 3: Cryptography

Basic concepts, symmetric and asymmetric encryption, RSA, digital signatures, hash functions, public key infrastructure (PKI).

Unit 4: Access Control and Authentication

Authentication factors, password-based systems, biometrics, access control models (DAC, MAC, RBAC).

Unit 5: Information Assurance and Cyber Laws

Confidentiality, integrity, availability (CIA triad), IT Act 2000, digital forensics basics, security policies, ethics and privacy.

Textbooks

1. William Stallings, *Cryptography and Network Security*, Pearson.
2. Nina Godbole & Sunit Belpure, *Cyber Security: Understanding Cyber Crimes*, Wiley.

Recommended Books

1. Behrouz Forouzan, *Cryptography and Network Security*, McGraw Hill.
2. Charles P. Pfleeger & Shari Lawrence Pfleeger, *Security in Computing*, Pearson.
3. Joseph Migga Kizza, *Computer Network Security and Cyber Ethics*, McFarland.

Murika
medina