FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

B.Sc.

Three year UG course

SYLLABUS

W.E.F. 2025-2026

DURGABAI DESHMUKH MAHILA SABHA (AMS)

ARTS AND SCIENCE COLLEGE FOR WOMEN

Autonomous NAAC Reaccredited

O.U. Road Hyderabad 500007

ANDHRA MAHILA SABHA
Arts & Science College for Women
(AUTONOMOUS)
OU CAMPUS , HYDERABAD NAAC RE-ACCREDITED

Scheme of Evaluation (BSC(MSCS)) for the Academic year 2025-26

BSc (MSCs) Computer Science I year I Semester

Subject Code	Subject	Hours / Week		Exam Duration	Marks			No Of Credits
		Theory	Practical		Sem	Internal	Total	
CSC 101 DSC	Programming in C	4 Hrs		3 Hrs	80	20	100	4
Practical's								
CSC 131	Programming in C		3 Hrs	3 Hrs	30	20	50	1

Head, Department of Computer Science
ANDHRA MAHILA SABHA
Arts & Science College for Women-Autonomous
O.U. Campus, Hyderabad - 500 007

BSc (MSCs) Computer Science I Year II Semester

Subject Code	Subject	Hours / Week		Exam Duration	Marks			No Of Credits
		Theory	Practical		Sem	Internal	Total	
Csc 151 DSC	Data Structures using C	4 Hrs		3 Hrs	80	20	100	4
Practicals								
Csc 181	Data Structures using C		3 Hrs	3 Hrs	30	20	50	1

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

Andhra Mahila Sabha Arts & Science College for Women
(AUTONOMOUS)
NAAC RE-ACCREDITED
O.U. CAMPUS, HYDERABAD-500 007

DEPARTMENT OF COMPUTER SCIENCE
I YEAR-I SEMESTER

2025-26

I Year I Semester
Paper – I
Programming in C

Subject Code: CSC 101	
Instruction	**4 Hrs/ Week**
Duration of the semester Examination	**3 hrs**
Duration of the Session Examination	**½ hrs**
Semester Examination	**80 Marks**
Session Examination	**20 Marks**
No of Credits	**4 Credits**

Course Objectives:

- **Cob1**: To introduce the basics of programming languages, focusing on the C programming language, and explore different ways of writing and designing algorithms.
- **Cob2**: To understand the key concepts such as variables, data types, operators, control structures, arrays and strings.
- **Cob3**: To emphasize problem-solving techniques using functions and pointers.
- **Cob4**: To learn the concept of user-defined data types and files.

Course Outcomes:

- **CO1**: Understand program structure and design algorithm.
- **CO2**: Develop basic programs by applying concepts such as control structures, arrays and strings.
- **CO3**: Implement functions, pointers and Dynamic Memory Allocation (DMA).
- **CO4**: Create user-defined data types and implement file operations.

Unit I

Programming Fundamentals:
Algorithms and Flow charts, Generation and classification of programming languages, Processes involved in program execution: compilation, interpretation, loading and linking.
Basics of C Programming:
Introduction to C programming language, Structure of a C program, C tokens, data types, variables, constants, operators, expression evaluation (precedence, associativity), type conversions in C.

Unit II

Input and Output:
Non-formatted and formatted input/output functions, Escape sequences and their usage in I/O.
Control Statements – Sequence statements, Selection statements: if, if-else, nested if, switch, conditional operators, Iterative statements: while, do-while and for.
Special control statements: goto, break, continue, return, exit.

Arrays and Strings:
One-dimensional arrays, Multidimensional arrays and character arrays.

Unit III

Functions:
Function definition, declaration and calling mechanisms, types of functions, ctype functions and string functions, call-by-value, call-by-reference.
Passing arrays to functions, recursion, inline functions. Scope and lifetime of variables, storage classes.

Pointers:
Introduction, address-of operator (&).
Uses of pointers, Pointer types: pointers and arrays, pointers and strings, pointer to pointer, array of pointers.
Dynamic memory allocation, malloc, calloc and free.

Unit IV

User-Defined Data Types:
Structures and unions: Definition, initialization, accessing members, arrays of structures, structures vs. unions, enumeration types (enum).

File handling:
Introduction, file operations, file functions: open, close, read and write. Working with text and binary files. Files of Records, Random Access to Files of Records.

Suggested Books

1. Reema Thareja, *"Programming in C"*, Oxford University Press, Second Edition, 2016.
2. Kernighan, B.W and Ritchie, D.M, *"The C Programming Language"*, Second Edition, Pearson Education, 2015.

Reference Books

1. Ivor Horton, *Beginning C*
2. Ashok Kamthane, *Programming in C*
3. Herbert Schildt, *The Complete Reference C*
4. Paul Deitel, Harvey Deitel, *C How to Program*
5. R.S. Bichkar, *"Programming with C"* University Press, 2024.
6. Byron S. Gottfried, *Theory and Problems of Programming with C*
7. Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*
8. B. A. Forouzan, R. F. Gilberg, *A Structured Programming Approach Using C*

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

I Year I Semester
Programming in C
Paper – I (Practical /laboratory)

Subject Code : CSC131	
Instruction	**: 3 Hrs/Week**
Duration of the semester Examination	**: 3 Hrs**
Marks for semester Examination	**: 50**
No of Credits	**: 1 Credit**

Course Objectives:

- **Cob1**: Develop fundamental programming skills in C by implementing conditional statements, loops, functions, and data structures for problem-solving.
- **Cob2**: Apply key C programming concepts such as arrays, pointers, strings, file handling, and recursion to build efficient algorithms for real-world applications.

Course Outcomes:

- **CO1**: Demonstrate the ability to write, debug, and execute C programs for solving mathematical and logical problems using control structures, functions, and data structures.
- **CO2**: Apply fundamental C programming concepts, including file handling, recursion, and memory management, to develop efficient solutions for computational tasks.

Lab Experiments:

1. Write a C program to input numbers and find the largest of two or three numbers using if statements and the conditional (ternary) operator (? :). Display the largest number.
2. Write a C program that takes an integer input and outputs the reversed number.
3. Write a C program to print all prime numbers between 2 and a given number *n*.
4. Write a C program to find the roots of a quadratic equation $ax^2 + bx + c = 0$.
5. Write a C program to print a triangle pattern of stars (*), where the number of lines is given by the user.
6. Write a C program to find the largest and smallest elements in an array of *n* numbers.
7. Write a C program to multiply two matrices of 3x3.
8. Write a C program to find the Greatest Common Divisor (GCD) of two numbers using both iteration and recursion.
9. Write a C program to demonstrate the use of different storage classes (auto, register, static, extern).
10. Write a C program to demonstrate the concepts of call-by-value and call-by-reference.
11. Write a C program that takes a string from the command-line arguments and counts the occurrence of each alphabet letter in the string.
12. Write a C program to demonstrate the usage of the enum data type.
13. Write a C program that demonstrates various string functions from the <string.h> library.
14. Write a C program that demonstrates structures and unions.
15. Write a C program that opens a file and counts the total number of characters in it.
16. Write a C program that copies content from an existing text file to a new file.

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

Head, Department of Computer Science
ANDHRA MAHILA SABHA
College for Women-Autonomous
Hyderabad - 500 007.

Andhra Mahila Sabha Arts & Science College for Women
(AUTONOMOUS)
NAAC RE-ACCREDITED
O.U. CAMPUS, HYDERABAD-500 007

DEPARTMENT OF COMPUTER SCIENCE
I YEAR-II SEMESTER

2025-26

I Year II Semester
Paper – II
Data Structures using C

Subject Code: CSC 151	
Instruction	**4 Hrs/ Week**
Duration of the semester Examination	**3 hrs**
Duration of the Session Examination	**½ hrs**
Semester Examination	**80 Marks**
Session Examination	**20 Marks**
No of Credits	**4 Credits**

Course Objectives:

- **Cob1**: To discuss the linear data structures and their applications.
- **Cob2**: To understand Queues, Linked list and Hashing Concepts.
- **Cob3**: To understand and implement trees and graphs with efficient traversal, searching, and optimization techniques.
- **Cob4**: Analyze and implement advanced searching and sorting techniques, including hashing and overflow handling, to optimize data organization and retrieval.

Course Outcomes:

- **CO1**: Understand and implement fundamental data structures, including arrays and stacks, for efficient data manipulation and expression evaluation.
- **CO2**: Apply linked lists, queues, and hashing techniques to optimize data storage, retrieval, and processing.
- **CO3**: Analyze and implement tree and graph structures, including traversal techniques and efficient searching strategies.
- **CO4**: Develop and optimize searching and sorting algorithms to enhance data organization and retrieval efficiency.

UNIT I
Introduction to Data structures: Definition, Types of Data structures.
Arrays: Arrays – ADT, ordered lists, Sparse matrices, representation of arrays.

Stacks: Stack ADT, Stacks using Arrays, Stacks using dynamic arrays,
Evaluation of Expressions – Evaluating Postfix Expression, Infix to Postfix expression, checking well-formed parenthesis, reversing a string.

UNIT II

Queues: Queues ADT, operations, Circular Queues, Applications.

Linked Lists: Singly Linked Lists and Chains, Linked Stacks and Queues, Polynomials, Operations for circularly linked lists, Equivalence Classes, Doubly Linked Lists.
Hashing: Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques.

UNIT III

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, Binary Search Trees (BST): Definition, Searching an element, Insertion into a BST, and Deletion from a BST, Efficient Binary Search Trees.
AVL Trees: Definition, Insert, search and delete operations.

Graphs: Graph Abstract Data Type, Elementary Graph operations, Graph Traversal Techniques - DFS and BFS, Minimum Cost Spanning Trees – Prim's and Kruskal's Algorithms.

UNIT IV

Searching and Sorting:

Sequential search, Binary search, Hash Tables: Hashing Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques.
Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Heap sort, List and Table Sorts.

Suggested Book

- **Horowitz E, Sahni S and Susan Anderson-Freed**, *Fundamentals of Data structures in C*, 2nd Edition (2008), Universities Press.

Reference Books

1. **Mark A Weiss**, *Data Structures and Algorithm Analysis in C*, Second Edition (2002), Pearson
2. **Kushwaha D. S and Misra A.K**, *Data Structures: A Programming Approach with C*, Second Edition (2014), PHI
3. **Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed**, *Fundamentals of Data Structures in C*, Second Edition (2008), Universities Press
4. **Tanenbaum A. M, Langsam Y, Augenstein M. J**, *Data Structures using C*, Second Edition (2008), Pearson
5. **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein**, *Introduction to Algorithms*, Third Edition (2009), MIT Press
6. **Chandan Banerjee and Atanu Das**, *Data Structures and Algorithms in C and PYTHON*, University Press, 2023
7. **Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum**, *Data Structures Using C and C++*, Second Edition (2009), PHI

P. V. Sudha

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

Ramana B

Head, Department of Computer Science
ANDHRA MAHILA SABHA
Arts & Science College for Women-Autonomous
O.U. Campus, Hyderabad - 500 095

I Year II Semester
Data Structures using C
Paper – II (Practical /laboratory)

Subject Code: CSC 181	
Instruction	**: 3 Hrs/Week**
Duration of the semester Examination	**: 3 Hrs**
Marks for semester Examination	**: 50**
No of Credits	**: 1 Credit**

Course Objectives:

Cob1: Develop and implement various data structures such as arrays, linked lists, stacks, queues, trees, and graphs for efficient data manipulation.
Cob2: Apply sorting, searching, and hashing techniques to solve computational problems effectively.

Course Outcomes:

CO1: Implement fundamental and advanced data structures, including arrays, linked lists, stacks, queues, trees, and graphs.
CO2: Apply efficient searching, sorting, and hashing techniques to solve computational problems.

Lab Experiments (Using C programming Language):

1. Implementation of Stacks and Queues using Arrays.
2. Implementation of Circular Queue.
3. Implementation of Infix to Postfix Conversion, Postfix Expression Evaluation.
4. Implementation of Singly Linked List
5. Implementation of Doubly Linked List.
6. Implementation of Circular Linked List.
7. Implementation of Stacks using Linked Lists
8. Implementation of Queues using Linked Lists.
9. Implementation of Linear search and Binary Search.
10. Implementation of Operations on Binary Tree
11. Implementation of Binary Search Tree.
12. Implementation of Traversal on Graphs.
13. Implementation of Selection, Bubble and Insertion Sort.
14. Implementation of Merge Sort.
15. Implementation of Quick Sort.

P.V. Sudha

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

Ramana-B

Head, Department of Computer Science
ANDHRA MAHILA SABHA
Arts & Science College for Women-Autonomous
O.U. Campus, Hyderabad - 500 007.