PARVATHANENI BRAHMAYYA
SIDDHARTHA COLLEGE OF ARTS & SCIENCE
Autonomous
Siddhartha Nagar, Vijayawada–520010
Re-accredited at 'A+' by the NAAC

Course Code				23DSMAP231			
Title of the Course				**Data structures using C Lab**			
Offered to:				B.Sc. Honours (Data Science)			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:		**2024-25**		**Semester:**		**3**	
Course Category:		**Major Lab**		**Course Relates to:**		**Local , Regional, National, Global**	
Year of Revision:		**--**		**Percentage:**		**--**	
Type of the Course:				**Skill Development**			
Crosscutting Issues of the Course :				**Professional Ethics**			
Pre-requisites, if any				**Algorithm Logic**			

Course Description:

This course covers fundamental concepts in data structures and algorithms. Topics include C program structure, control structures, array data structures, algorithm analysis, sorting algorithms, stacks, queues, dynamic arrays, linked lists, trees, and graphs. Students will learn implementation, properties, functions, and traversal techniques, emphasizing efficiency and practical problem-solving.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	To choose an appropriate data structure as applied to a specified problem.
2	To use various techniques for representation of the data in the real world.
3	To understand applications using data structures.
4	To develop graph data structures, traversal algorithm.
5	To test the logical ability for solving problems.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Remember C program structure, control structures, and data types.	**K1**	**PO1, PO5, PO6, PO7**	**PSO1, PSO2**
CO2	Understand arrays, Big O notation, time and space complexity.	**K2**	**PO1,PO5,PO6 , PO7**	**PSO1, PSO2**
CO3	Apply sorting algorithms using arrays in practical scenarios.	**K3**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**
CO4	Analyze and compare stacks, queues, dynamic arrays, linked lists.	**K4**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**
CO5	Evaluate tree and graph data structures, traversal algorithm efficiency.	**K5**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**2**	**2**	**2**	**2**	**1**
CO2	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO3	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO4	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO5	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of a **Data structures using C** course, providing hands-on practice with (Turbo C++)

Unit 1: [Introduction to Data Structures and Basic Concepts](6Hrs)

Lab 1:

4. Write a C program to find Compound Interest.
5. Write a C Program to find Roots of a quadratic equations with nature of roots.
6. Write a C program to find factorial of a given number.

- **Dataset** (web link) **/ Experiment**: Explore basic programming constructs in C.
- **Tasks:**
  3. **Basic Program:** Write a C program to calculate Simple Interest.
  4. **Control Structures:** Write a C program using if, while, for, and do-while loops.

Lab 2:

4. Write a C program to read n numbers and count number of +ve,-ve, zeros.
5. Write a C program to find sum of digits of a given number.
6. Write a C program to find Armstrong numbers between 1-1000.

- **Dataset (web link) / Experiment:** Perform basic operations on using while,for in C.
- **Tasks:**
  4. **Count Even/Odd Elements:** Write a C program to count even and odd elements of 10 integers.
  5. **Strong Number:** Write a C program to find given number is strong or not.

Lab 3:

1. Write a C program to read elements and display in matrix.
2. Write a C program to display the identity matrix of given size.
6. Write a C program to find multiplication of two matrices.

- **Dataset (web link) / Experiment:** Perform basic operations on using while, for in C.
- **Tasks:**
  1. **Transpose of Matrix:** Write a C program to transpose a matrix.
  2. **Sum of matrices:** Write a C program to find sum of two matrices.

Unit 2: [Sorting Algorithms](6Hrs)

Lab 4:

3. Write a C program to implement the Bubble sort.
4. Write a C program to implement the Insertion sort.

- **Dataset** (web link) / **Experiment**: Implement and understand basic sorting algorithms.
- **Tasks:**
  3. **Bubble Sort:** Write a C program to sort an array using Bubble Sort.
  4. **Insertion Sort:** Write a C program to sort an array using Insertion Sort.

Lab 5:

3. Write a C program to implement the Selection sort.
4. Write a C program to implement the Quick sort.

- **Dataset (web link) / Experiment:** Implement advanced sorting algorithms.
- **Tasks:**

3. **Selection Sort:** Write a C program to sort an array using Selection Sort.
4. **Quick Sort:** Write a C program to sort an array using Quick Sort.

Unit 3: [Stack and Queue Data Structures](6Hrs)

Lab 6:

2. Write a C program to implement the stack operations.

- **Dataset** (web link) / **Experiment**: Implement basic stack operations (`push, pop, and  peek`) to understand the Last-In-First-Out (LIFO) principle in C.
- **Tasks:**
  2. **Stack Operations:** Write a C program to perform push, pop, display operations on a stack.

Lab 7:

2. Write a C program to implement the queue operations

- **Dataset (web link) / Experiment:** Implement basic queue operations (`enqueue, dequeue`) to understand the First-In-First-Out (FIFO) principle in C.
- **Tasks:**
  2. **Queue Operations:** Write a C program to perform enqueue, dequeue, and display operations on a queue.

Unit 4: [Dynamic Arrays and Linked Lists](6Hrs)

Lab 8:

2. Write a C program to implement the infix to postfix expression

- **Dataset** (web link) / **Experiment**: Convert infix expressions to postfix in C.
- **Tasks:**
  2. **Infix to Postfix Conversion:**
     Write a C program to convert infix expressions to postfix.

Unit 5: [Trees and Graphs] (6Hrs)

Lab 9:

2. Write a C program to implement the Binary Search Tree operations.

- **Dataset** (web link) / **Experiment**:Implement Binary Search Tree (BST) operations to manage hierarchical data in C.
- **Tasks:**

BST Operations: Write a C program to perform insert, delete, and search operations on a Binary Search Tree (BST).

Lab 10:

3. Write a C program to implement the BFS traversal algorithm for a graph.
4. Write a C program to implement the DFS traversal algorithm for a graph.

- **Dataset** (web link) / **Experiment**:Implement BFS and DFS traversal algorithms for exploring graphs in C.
- **Tasks:**
  3. **BFS Traversal:** Write a C program to perform Breadth-First Search (BFS) on a graph.
  4. **DFS Traversal:**Write a C program to perform Depth-First Search (DFS) on a graph.

Lab Manual:

2. "Data Structures and Algorithms in Java" by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser , Third edition.

References:

3. "Data Structures Using C and C++" by Yedidyah Langsam, Moshe Augenstein, Aaron M. Tenenbaum.
4. Reema Thareja, Data Structures Using C, Oxford University Press Publishers,2023.

Lab- Question Paper Pattern

23DSMAP231: Data Structures Using C Lab

Offered to: B.Sc. Honours (Data Science)

Max. Marks: 50 **Max. Time: 3Hrs**

Pass. Min: 20

(A)	**Evaluation Procedure**	**35 Marks**
I	Experiments (Exam & Execution)	**30 Marks**
II	Viva	**3 Marks**
III	Record	**2 Marks**
(B)	**CONTINUOUS ASSESMENT(Internal)**	**15 MARKS**

15 marks for the continuous assessment (Day to day work in the laboratory shall be evaluated for 15 marks by the concerned laboratory teacher based on the regularity/ record/viva). Laboratory teachers are mandated to ensure that every student completes 80%-90% of the lab assessments.

TOTAL: (A)+(B) = **50 MARKS**