



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

II Year – I Semester				
	L	T	P	C
	0	0	3	1.5
<b>DATA STRUCTURES THROUGH C++ LAB</b>				

**Course Objectives:**

The objective of this lab is to

- Demonstrate procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
- Understand dynamic memory management techniques using pointers, constructors, destructors, etc
- Demonstrate the concept of function overloading, operator overloading, virtual functions and polymorphism, inheritance.
- Demonstrate the different data structures implementation.

**Course Outcomes:**

By the end of this lab the student is able to

- Apply the various OOPs concepts with the help of programs.
- Use basic data structures such as arrays and linked list.
- Programs to demonstrate fundamental algorithmic problems including Tree Traversals, Graph traversals, and shortest paths.
- Use various searching and sorting algorithms.

**Exercise -1 (Classes Objects)**

Create a Distance class with:

- feet and inches as data members
  - member function to input distance
  - member function to output distance
  - member function to add two distance objects
1. Write a main function to create objects of DISTANCE class. Input two distances and output the sum.
  2. Write a C++ Program to illustrate the use of Constructors and Destructors (use the above program.)
  3. Write a program for illustrating function overloading in adding the distance between objects (use the above problem)

**Exercise – 2 (Access)**

Write a program for illustrating Access Specifiers public, private, protected

1. Write a program implementing Friend Function
2. Write a program to illustrate this pointer
3. Write a Program to illustrate pointer to a class

**Exercise -3 (Operator Overloading)**

1. Write a program to Overload Unary, and Binary Operators as Member Function, and Non Member Function.
  1. Unary operator as member function
  2. Binary operator as non member function
2. Write a c ++ program to implement the overloading assignment = operator

**Exercise -4 (Inheritance)**

1. Write C++ Programs and incorporating various forms of Inheritance
  - i) Single Inheritance
  - ii) Hierarchical Inheritance



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

- iii) Multiple Inheritances
- iv) Multi-level inheritance
- v) Hybrid inheritance

2. Also illustrate the order of execution of constructors and destructors in inheritance

**Exercise -5(Templates, Exception Handling)**

1. a) Write a C++ Program to illustrate template class
2. b) Write a Program to illustrate member function templates
3. c) Write a Program for Exception Handling Divide by zero
4. d) Write a Program to rethrow an Exception

**Exercise -6 (Searching)**

Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.

b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.

**Exercise -7 (Sorting-I)**

- a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order
- b) Write C program that implement Quick sort, to sort a given list of integers in ascending order
- c) Write C program that implement Insertion sort, to sort a given list of integers in ascending order

**Exercise -8(Sorting-II)**

- a) Write C program that implement radix sort, to sort a given list of integers in ascending order
- b) Write C program that implement merge sort, to sort a given list of integers in ascending order

**Exercise -9(Singly Linked List)**

- a) Write a C program that uses functions to create a singly linked list
- b) Write a C program that uses functions to perform insertion operation on a singly linked list
- c) Write a C program that uses functions to perform deletion operation on a singly linked list
- d) Write a C program to reverse elements of a single linked list.

**Exercise -10(Queue)**

- a) Write C program that implement Queue (its operations) using arrays.
- b) Write C program that implement Queue (its operations) using linked lists

**Exercise -11(Stack)**

- a) Write C program that implement stack (its operations) using arrays
- b) Write C program that implement stack (its operations) using Linked list
- c) Write a C program that uses Stack operations to evaluate postfix expression

**Exercise -12(Binary Search Tree)**

- a) Write a C program to Create a BST
- b) Write a C program to insert a node into a BST.
- c) Write a C program to delete a node from a BST.
- d) Write a recursive C program for traversing a binary tree in preorder, inorder and postorder.