**National University of Computer and Emerging Sciences**



**Laboratory Manual**

*for*

# Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Mr. Uzair Naqvi |
| Lab Instructor | Mr. Durraiz Waseem |
| Semester | Fall 2024 |

# Department of Computer Science

FAST-NU, Lahore, Pakistan

**Objectives:**

In this lab, students will practice:

1. Binary Trees
2. Binary Search Trees (BST)

**Problem 1: Binary Tree Search and Insertion**

Implement a Binary Trees class with the following operations:

a) **insert( int key):** Inserts a new node with the given key into the Binary Tree.

b) **search( int key):** Returns true if the Binary Tree contains a node with the given key, or false if not.

**Problem 2: Binary Tree Inorder Traversal**

Implement a function to perform an inorder traversal of a Binary Tree and return the elements in sorted order.

**Problem 3: Binary Tree Deletion**

Implement a function to delete a node with a given key from a Binary Tree. Handle cases when the node to be deleted has zero, one, or two children.

**Problem 4: BST Insertion and Deletion**

Implement a Binary Search Tree (BST) class with the following operations:

a) **insert(int key):** Inserts a new node with the given key into the BST.

b) **delete(int key):** deletes a node with a given key from a BST. Handle cases when the node to be deleted has zero, one, or two children.

**Problem 5: BST Range Search**

Implement a function to find all nodes in a BST with keys falling within a specified range [low, high].

Hint: find minimum and maximum nodes all over the tree.

**Problem 6: Second Largest Element in BST**

Write a function to find and return the second-largest element in a BST. The second largest element is the one that has the second-highest value.

**Problem 7: BST Level Order Traversal**

Implement a function to perform a level-order traversal of a BST and return the elements level by level.