Lab 12: Data Structures and Algorithms

**Topic: BST Continued**

# Objective

* Working on the Binary Search Trees

|  |  |
| --- | --- |
| **Name:** |  |
| **Registration No.:** |  |

**Task 1: portal Quiz 12 – first 60 minutes**

Time Allowed: 60 minutes [extra time will not be given] Total Marks: **(10)**

**Question:**

Write down the recursive code to INSERT a value in a binary search tree. Note that the prototype of the recursive function must be:

**void insertRec(Node<T>\*p, Node<T>\*newNode)**

where \*newNode is a pointer to the node that has to be added/inserted. Memory has to be allocated to it in the wrapper function.

Other than insert, code of post-order traversal is also required. You are required to create the complete code (complete class BST and struct Node). Main program is not required, it has been written below for your convenience:

int main()

{

BST<int> tree;

tree.INSERT(100); //**void INSERT(T)** is a wrapper function

tree.INSERT(50);

tree.INSERT(75);

tree.INSERT(150);

tree.postorder(); //post-order traversal also required

return 0;

}

Lab 11: Data Structures and Algorithms

**Topic: BST Continued**

## Task 2: Lab Task – remaining time of the lab

In the code that is already present on LMS, add the complete code for:

1. Deletion – 2 child case and root cases have to be added. Another addition is required: the case to check if the value to be deleted is NOT present in the BST
2. Searching in a BST. Function prototype in the class must be: **bool search (T value);**

Test the task:

1. Delete a value that is not present in a BST
2. Delete the value present at the root in such a way that:
3. Root has no children
4. Root has only one leaf child
5. Root has only one NON-LEAF child
6. Root has two leaf children
7. Root had two NON-LEAF children

In the main program, create a different object of the BST to test each of the cases above. Add proper comments and cout statements so that it is easy for the lab instructors to grade.

**Task 3: Lab Task – remaining time of the lab**

In the destructor, write down the complete code to DESTROY the BST. You will be using the reserved word delete. Do test your code for the following main program:

int main()

{

BST<int> test;

return 0;

}