# University of Central Punjab Faculty of Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |  |
| --- | --- | --- |
| **Lab 13** | |  |
| **Topic** | * Trees ADT * Binary Search Tree (BST) * Recursion |
| **Objective** | The basic purpose of this lab is to practice the insertion,deletion,searching of BST and traversal of BST |

**Instructions:**

* Indent your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

# void main() is not allowed. Use int main()

* **You have to work in multiple files. i.e separate .h and .cpp files**
* **You are not allowed to use any built-in functions**

# You are required to follow the naming conventions as follow:

* + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

# Students are required to complete the following tasks in lab timings.

**Task 1**

* Create Struct **Node** which is having

***Attributes:***

* **Type data;**
* **Node<Type>\* left;**
* **Node<Type>\* right**
* Create abstract class named as **BST**

***Attributes of BST:***

* **Node<Type>\* root;**

***Functions:***

* **Virtual void insert\_node(Type value) =0; //wrapper function**

This function inserts a node in the binary search tree

* **Virtual void Print\_inOrder()= 0; //wrapper function**

This function traverse and prints all the values of binary search tree the tree in order method

* **Virtual void Print\_PostOrder()= 0; //wrapper function**

This function traverse and prints all the values of binary search tree the tree post order method

* **Virtual void Print\_PreOrder()= 0; //wrapper function**

This function traverse and prints all the values of binary search tree the tree pre order method

Create **constructor** for this class

Now you have to make a derived class named as **myBST** andimplement the above functions

After Implementation of the functions in **myBST** create menu based program to perform the following operations .:

1. **Insert data**
2. **Print inOrder**
3. **Print postOrder**
4. **Print preOrder**
5. **Exit**

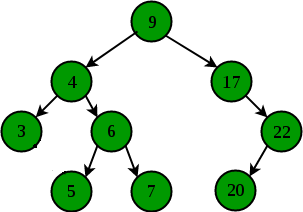
**Task 2 Find the closest element in BST**

Create Given a BST and a target node **K**. The task is to find the node with a minimum absolute difference with the given target value K.

EXAMPLES:

***Input :****k = 4****Output:****4*

***Input :****k = 18****Output:****17*



**Task 3 Print Common Nodes in Two Binary Search Trees**

Given two Binary Search Trees, find common nodes in them. In other words, find the intersection of two BSTs.

**Example:**

***Input:****root1:  
                   5  
             /          \  
         1              10  
     /      \          /  
  0        4       7  
                      \  
                       9*

*root2:    10  
           /       \  
        7         20  
    /      \  
4          9****Output:****4 7 9 10*