# University of Central Punjab Faculty of Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |  |
| --- | --- | --- |
| **Lab 12** | |  |
| **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**

* 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 delete\_node(Type value) =0; //wrapper function**

This function delete a node in the binary search tree

* **Virtual Type find\_min() =0; //wrapper function**

This function to find a min node in the binary search tree

* **Virtual Type find\_max() =0; //wrapper function**

This function to find a max 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. **Delete data**
3. **Find min node in BST**
4. **Find max node in BST**
5. **Print inOrder**
6. **Print postOrder**
7. **Print preorder**
8. **Exit**

**Task 3**

* 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 delete\_node(Type value) =0; //wrapper function**

This function delete a node in the binary search tree

* **Virtual Type find\_min() =0; //wrapper function**

This function to find a min node in the binary search tree

* **Virtual Type find\_max() =0; //wrapper function**

This function to find a max node in the binary search tree

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

This function search 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. **Delete data**
3. **Search data**
4. **Print inOrder**
5. **Print postOrder**
6. **Print preorder**
7. **Exit**