University of Central Punjab

## Faculty of Information Technology

**Data Structures and Algorithms Spring 2023**

|  |  |  |
| --- | --- | --- |
| **Lab 9** | |  |
| **Topic** | * Trees ADT * Binary Search Tree (BST) * Recursion |
| **Objective** | The basic purpose of this lab is to practice the insertion of BST and traversal of BST |

**Instructions:**

* Indent your code.
* Comment 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 system("pause")

* **You are not allowed to use any built-in functions**

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

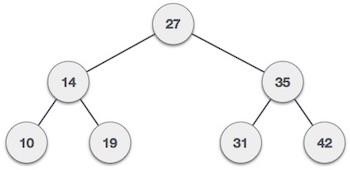
* + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

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

**Binary Search Tree:**

A Binary Search Tree (BST) is a tree in which all the nodes follow the below-mentioned properties −

* The value of the key of the left sub-tree is less than the value of its parent (root) node's key.
* The value of the key of the right sub-tree is greater than or equal to the value of its parent (root) node's key.



1. Inorder

(Left, Root, Right):10 14 19 27 31 35 42

1. Preorder (Root, Left, and Right): 27 14 10 19 35 31 42
2. Postorder (Left, Right, and Root): 10 19 14 31 42 35 27

# Task 1

Create a C++ generic abstract class Node

**Attributes:**

1. Type data;
2. Node \* left\_child;
3. Node \* right\_child;

**Functions:**

* + Write parameterized constructor with default arguments for the above class.
  + Write Copy constructor for the above class.
  + Write Destructor for the above class.

# Task 2

Create an abstract class tree using the node class created in Task 1:

**Attributes:**

1. Node \*Root

**Functions:**

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

This functions inserts a node in the binary search tree

## Virtual void Print\_inOrder()= 0;

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

## Virtual void Print\_PostOrder()= 0;

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

## Virtual void Print\_PreOrder()= 0;

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

* Write parameterized constructor with default arguments for the above class.
  + Write Copy constructor for the above class.
  + Write Destructor for the above class.

# Task 3

Create a class BST by using the abstract class created in Task 2. Override already declared virtual method “Insert\_node()” according to BST and also implement other virtual methods.

**Attributes:**

1. Node \*Root

**Functions:**

* **Bool is\_Bst\_Empty( )**

This functions check whether the binary search tree is empty or not

* Write parameterized constructor with default arguments for the above class.
  + Write Copy constructor for the above class.
  + Write Destructor for the above class.
* **delete\_Leaf\_Node (Node\* root, int k )**
* **delete \_Node\_One\_Child (Node\* root, int k )**
* **delete \_Node\_Two\_Child (Node\* root, int k )**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***