**Overview:**

This program implements a binary search tree. The binary search tree is a generic class which allows any class to be inserted in it. Values that are added to the tree are sorted. The contents of the tree can be displayed with three methods: preorder, inorder, and postorder.

**Processing logic:**

Create a binary search tree by setting the first value as root

Add element to the tree

Check if the tree is empty

Check if the tree already contains that element

If both are false, find a null node to insert the element into so that it is sorted

Print the inorder sorted values

Print the preorder sorted values

Print the postorder sorted values

**Data:**

The binary search tree class and its node class can contain generic data types. Any data type or class can be used.

**Components:**

binarySearchTree.h

Define generic node class, which contains an element of generic type and pointers to left and right nodes. Defines the variables and functions for the binarySearchTree, which is listed in detail in the next paragraph. Briefly, the binarySearchTree class has two nodes, the current node and the root node, and the number of nodes in the tree.

binarySearchTree.cpp

Contains definitions for instantiating the class, finding similar elements in the tree, checking if the tree is empty, getting and setting the current node, getting the root node, adding elements to the tree, destroying the tree, getting the size of the tree, and listing the contents of the tree using preorder, inorder, and postorder sorting.

|  |
| --- |
| **binarySearchTree** |
| -sz: int |
| -current: node<T> |
| -root: node<T> |
| +setCurrent(node<T>): void |
| +getCurrent(): \*node<T> |
| +getRoot(): \*node<T> |
| +add(T): void |
| +find(T): \*node<T> |
| +destroy(\*node<T>): void |
| +preorder(node<T>): void |
| +inorder(node<T>): void |
| +postorder(node<T>): void |
| +size(): int |