Name – Yash Daga Roll Number – 20BCE7323

1.Write a program to implement standard BST operations(insert,delete,search) and also write a method height() to return height of a BST.

Code:

```
class BinarySearchTree {
   class Node
   {
      int key;
      Node left, right;

      public Node(int item)
      {
         key = item;
         left = right = null;
      }
   }
    Node root;
   BinarySearchTree()
   {
       root = null;
   }

   void insert(int key)
   {
       root = insertRec(root, key);
   }
```

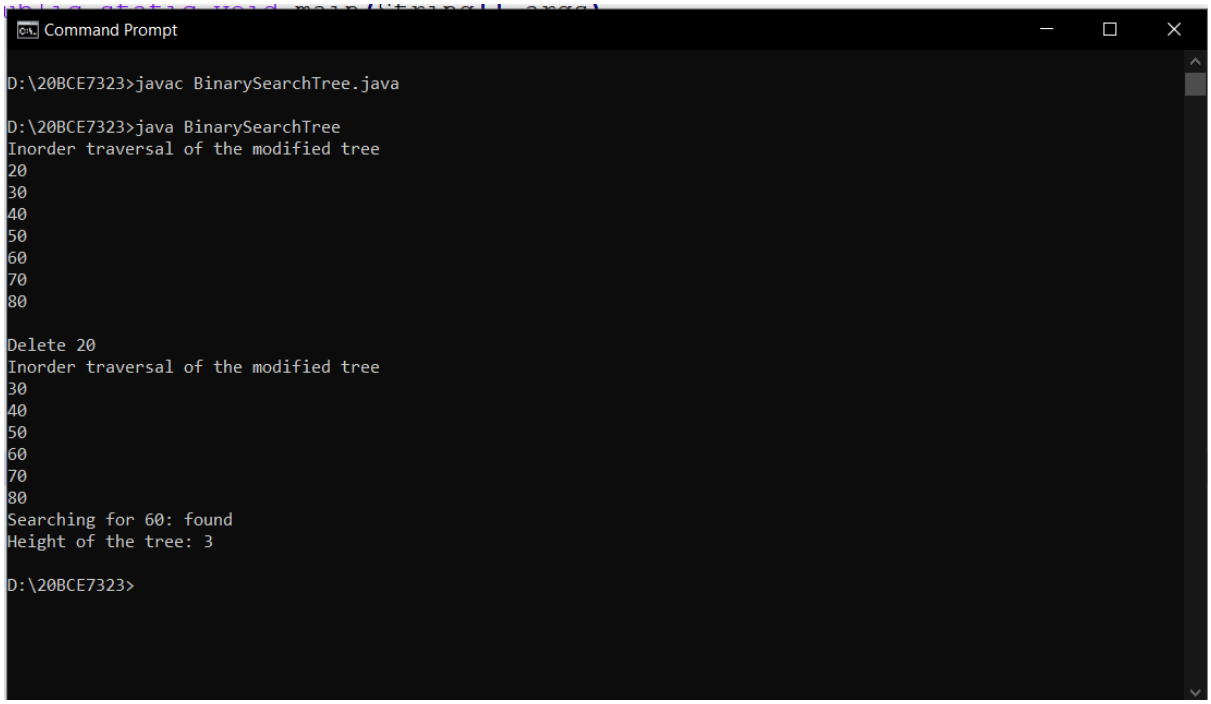
```
Node insertRec(Node root, int key)
{

   if (root == null)
   {
      root = new Node(key);
      return root;
   }
    if (key < root.key)
      root.left = insertRec(root.left, key);
   else if (key > root.key)
      root.right = insertRec(root.right, key);

   return root;
}
 void inorder()
{
    inorderRec(root);
}
 void inorderRec(Node root)
{
   if (root != null) {
      inorderRec(root.left);
      System.out.println(root.key);
      inorderRec(root.right);
   }
```

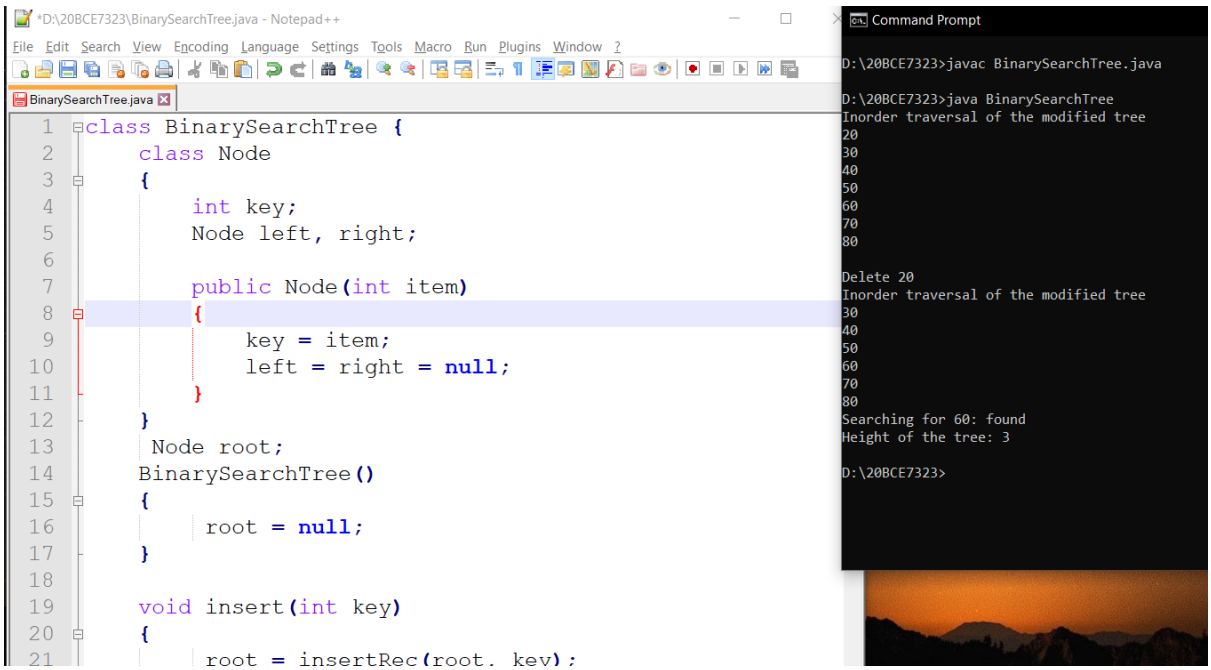
```
}
void deleteKey(int key) { root = deleteRec(root, key); }
Node deleteRec(Node root, int key)
{
   if (root == null)
      return root;
    if (key < root.key)
      root.left = deleteRec(root.left, key);
   else if (key > root.key)
      root.right = deleteRec(root.right, key);
    else {
      if (root.left == null)
         return root.right;
      else if (root.right == null)
         return root.left;
       root.key = minValue(root.right);
       root.right = deleteRec(root.right, root.key);
   }

   return root;
}
      int minValue(Node root)
{
   int minv = root.key;
   while (root.left != null)
   {
```

```
        minv = root.left.key;
        root = root.left;
    }
    return minv;
}

       Node search(Node root, int key)
{
if (root==null || root.key==key)
    return root;
 if (root.key < key)
   return search(root.right, key);
 return search(root.left, key);
}

int maxHeight(Node node)
{
    if (node == null)
        return 0;
    else
    {
        int lHeight = maxHeight(node.left);
        int rHeight = maxHeight(node.right);
        if (lHeight > rHeight)
            return (lHeight + 1);
         else
```

```
            return (rHeight + 1);
        }
          }
    public static void main(String[] args)
    {
        BinarySearchTree tree = new BinarySearchTree();
        tree.insert(50);
        tree.insert(30);
        tree.insert(20);
        tree.insert(40);
        tree.insert(70);
        tree.insert(60);
        tree.insert(80);
                System.out.println(
           "Inorder traversal of the modified tree");
        tree.inorder();
                System.out.println("\nDelete 20");
        tree.deleteKey(20);
        System.out.println(
           "Inorder traversal of the modified tree");
        tree.inorder();
                Node found = tree.search(tree.root, 60);
                if(found != null) {
                        System.out.println("Searching for 60: found");
                } else {
                        System.out.println("Searching for 60: not found");
```

```
                }
                System.out.println("Height of the tree: " +
tree.maxHeight(tree.root));
    }
}
```


Command Prompt
D:\20BCE7323>javac BinarySearchTree.java
D:\20BCE7323>java BinarySearchTree
Inorder traversal of the modified tree
20
30
40
50
60
70
80
Delete 20
Inorder traversal of the modified tree
30
40
50
60
70
80
Searching for 60: found
Height of the tree: 3
D:\20BCE7323>


*D:\20BCE7323\BinarySearchTree.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
BinarySearchTree.java
class BinarySearchTree {
class Node
{
int key;
Node left, right;
public Node(int item)
{
key = item;
left = right = null;
}
}
Node root;
BinarySearchTree()
{
root = null;
}
void insert(int key)
{
root = insertRec(root, key);
Command Prompt
D:\20BCE7323>javac BinarySearchTree.java
D:\20BCE7323>java BinarySearchTree
Inorder traversal of the modified tree
20
30
40
50
60
70
80
Delete 20
Inorder traversal of the modified tree
30
40
50
60
70
80
Searching for 60: found
Height of the tree: 3
D:\20BCE7323>