

LAB EXAM

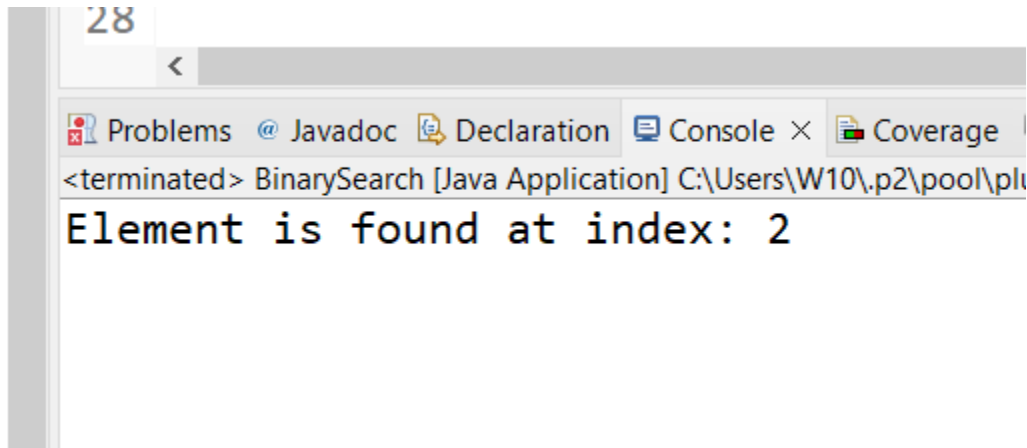
DATA STRUCTURE AND ALGORITHMS

1. Write a Java program to

a. Perform binary search operation

```
package com.Binaryprgm;

public class BinarySearch {
    public static void binarySearch(int arr[], int first,
int last, int key){
        int mid = (first + last)/2;
        while( first <= last ){
            if ( arr[mid] < key ){
                first = mid + 1;
            }else if ( arr[mid] == key ){
                System.out.println("Element is found at
index: " + mid);
                break;
            }else{
                last = mid - 1;
            }
            mid = (first + last)/2;
        }
        if ( first > last ){
            System.out.println("Element is not found!");
        }
    }
    public static void main(String args[]){
        int arr[] = {10,20,30,40,50};
        int key = 30;
        int last=arr.length-1;
        binarySearch(arr,0,last,key);
    }
}
```



b. Execute tree traversal in postorder

```
package com.postorder;
```

```
public class Node {  
    public int value;  
    public Node left, right;  
  
    public Node(int element)  
    {  
        value = element;  
        left = right = null;  
    }  
}
```

```
package com.postorder;
```

```
public class PostorderTraversal {  
    Node root;  
  
    PostorderTraversal() { root = null; }  
  
    void traversePostorder(Node node)  
    {
```

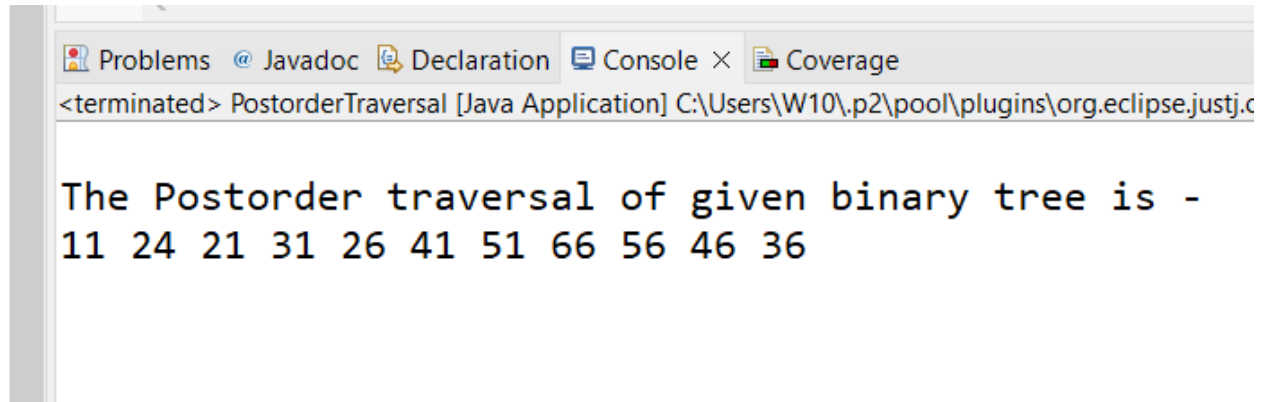
```
        if (node == null)
            return;
        traversePostorder(node.left);
        traversePostorder(node.right);
        System.out.print(node.value + " ");
    }

    void traversePostorder() { traversePostorder(root); }

    public static void main(String args[])
    {
        PostorderTraversal pt = new PostorderTraversal();
        pt.root = new Node(36);
        pt.root.left = new Node(26);
        pt.root.right = new Node(46);
        pt.root.left.left = new Node(21);
        pt.root.left.right = new Node(31);
        pt.root.left.left.left = new Node(11);
        pt.root.left.left.right = new Node(24);
        pt.root.right.left = new Node(41);
        pt.root.right.right = new Node(56);
        pt.root.right.right.left = new Node(51);
        pt.root.right.right.right = new Node(66);

        System.out.println();
        System.out.println("The Postorder traversal of given
binary tree is - ");
        pt.traversePostorder();
        System.out.println();
    }
}
```

Vishal Sambre
220960920075



The screenshot shows the Eclipse IDE's console window. The title bar at the top includes tabs for 'Problems', 'Javadoc', 'Declaration', 'Console', and 'Coverage'. The 'Console' tab is active, displaying the output of a Java application named 'PostorderTraversal'. The output text is: 'The Postorder traversal of given binary tree is - 11 24 21 31 26 41 51 66 56 46 36'. The console path is visible as 'C:\Users\W10\p2\pool\plugins\org.eclipse.justj.c'.

```
<terminated> PostorderTraversal [Java Application] C:\Users\W10\p2\pool\plugins\org.eclipse.justj.c  
  
The Postorder traversal of given binary tree is -  
11 24 21 31 26 41 51 66 56 46 36
```