Assignment No.1
Roll No:2460

```
import java.util.Stack;
class Node {
    Node left;
    int data;
    Node right;
    Node(){
    }
    Node(int data){
        left=null;
        right=null;
        this.data=data;
    }
}

class Traversals{
    public static Node Insert(Node root,int val){
        if(root==null){
         root= new Node(val);
       }
       else{

          if(val<root.data){
          root.left=  Insert(root.left,val);
          }
          else{
            root.right=Insert(root.right,val);
          }
        }
        return root;
    }

   public static  void inorder(Node root){
    if(root!=null){
          inorder(root.left);
          System.out.println(root.data);
          inorder(root.right);
      }
    }

    public static void preorder(Node root){
      if(root!=null){
        System.out.println(root.data);
        preorder(root.left);
        preorder(root.right);
      }
```

```
}

public static void postorder(Node root){
  if(root!=null){
    postorder(root.left);
    postorder(root.right);
    System.out.println(root.data);
  }
}

public static  void inorderN(Node root){
Stack<Node> stack=new Stack<Node>();
while(root!=null||!stack.empty()){
  while(root!=null){
    stack.push(root);
    root=root.left;
  }
  if(!stack.empty()){
    root=stack.peek();
    System.out.println(root.data);
   stack.pop();
   root=root.right;
  }
}
  }

  public static void preorderN(Node root){
    Stack<Node> stack=new Stack<Node>();
    while(root!=null||!stack.empty()){
      while(root!=null){
        System.out.println(root.data);
        stack.push(root);
        root=root.left;
      }
      if(!stack.empty()){
        root=stack.peek();
       stack.pop();
       root=root.right;
      }
    }
  }

  public static void postorderN(Node root){
    Stack<Node> stack1=new Stack<Node>();
    Stack<Character>stack2=new Stack<Character>();

    while(root!=null||!stack1.empty()){
      while(root!=null){
```

```
        stack1.push(root);
        stack2.push('L');
        root=root.left;
      }
      if(!stack1.empty()){
        root=stack1.peek();
        stack1.pop();
       Character c= stack2.pop();
        if(c=='L'){
      stack1.push(root);
      stack2.push('R');
      root=root.right;
        }
        else{
          System.out.println(root.data);
          root=null;

        }

    }
  }
}

public static void main(String[] args) {

    Node root=null;

     root=Insert(root,12);
     root=Insert(root,2);
     root=Insert(root,112);
     root=Insert(root,34);
     root=Insert(root,78);
     root=Insert(root,9);
     System.out.println("Recursive Traversal ");
     System.out.println("Inorder Traversal ");
    inorder(root);
    System.out.println("Preorder Traversal");
    preorder(root);
    System.out.println("Postorder Traversal");
    postorder(root);

    System.out.println("Non Recursive Traversal ");
    System.out.println("Inorder Traversal ");
   inorderN(root);
   System.out.println("Preorder Traversal");
   preorderN(root);
   System.out.println("Postorder Traversal");
```

```
        postorderN(root);

    }
}
```