DSA Lab Exam

1.

a. Perform inorder tree traversal

Code:

```
package com.inorder.pojo;

public class Node {

    public int key;
    public Node left;
    public Node right;

    public Node(int key) {
        this.key = key;
        this.left = null;
        this.right = null;
    }

}
```

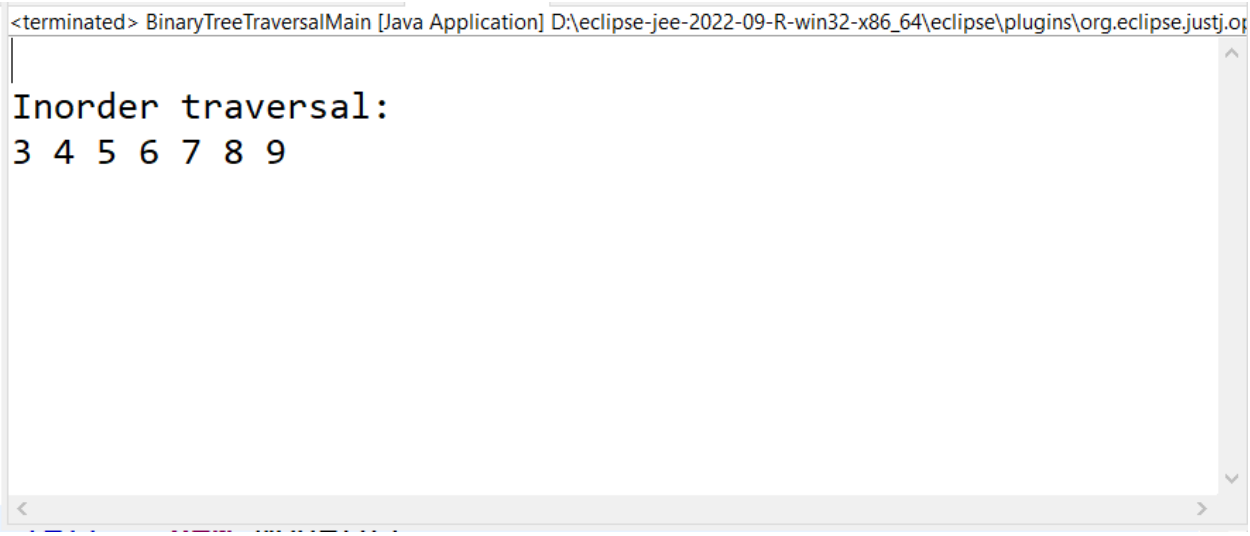
```
package com.inorder.pojo;

public class BinaryTreeTraversal {
public Node root;

    public BinaryTreeTraversal() {
        this.root = null;
    }
```

```
    public void printInorder(Node node) {
        if(node == null) {
            return;
        }

        printInorder(node.left);
        System.out.print(node.key + " ");
        printInorder(node.right);
    }


    public void printInorder() {

        printInorder(root);
    }

}
```

```
package com.inorder.main;

import com.inorder.pojo.BinaryTreeTraversal;
import com.inorder.pojo.Node;

public class BinaryTreeTraversalMain {

    public static void main(String[] args) {
BinaryTreeTraversal bt = new BinaryTreeTraversal();

        bt.root = new Node(6);
```

```
        bt.root.left = new Node(4);
        bt.root.right = new Node(8);
        bt.root.left.left = new Node(3);
        bt.root.left.right = new Node(5);
        bt.root.right.left = new Node(7);
        bt.root.right.right = new Node(9);


        System.out.println();
        System.out.println("Inorder traversal: ");
        bt.printInorder();
    }

}
```

Output:



b. Implement stack using array

Code:

```
package com.inorder.main;

import java.util.Scanner;

public class StackImp {


    int top;
    int maxsize = 10;
    int[] arr = new int[maxsize];


    boolean isEmpty()
    {
        return (top < 0);
    }
    StackImp()
    {
        top = -1;
    }
    boolean push (Scanner sc)
    {
        if(top == maxsize-1)
        {
            System.out.println("Overflow !!");
            return false;
        }
        else
        {
            System.out.println("Enter Value");
            int val = sc.nextInt();
            top++;
            arr[top]=val;
```

```
            System.out.println("Item pushed");
            return true;
        }
    }
    boolean pop ()
    {
        if (top == -1)
        {
            System.out.println("Underflow !!");
            return false;
        }
        else
        {
            top --;
            System.out.println("Item popped
successfully");
            return true;
        }
    }
    void display ()
    {
        System.out.println("Elements are: ");
        for(int i = top; i>=0;i--)
        {
            System.out.println(arr[i]);
        }
    }

public static void main(String[] args) {
    int choice=0;
    Scanner sc = new Scanner(System.in);
    StackImp s = new StackImp();
    while(choice != 4)
    {
```

```
        System.out.println("Chose one from the below
options...\n");

System.out.println("1.Push\n2.Pop\n3.Show\n4.Exit");
        System.out.println("Enter your choice \n");
        choice = sc.nextInt();
        switch(choice)
        {
            case 1:
            {
                s.push(sc);
                break;
            }
            case 2:
            {
                s.pop();
                break;
            }
            case 3:
            {
                s.display();
                break;
            }
            case 4:
            {
                System.out.println("Exiting....");
                System.exit(0);
                break;
            }
            default:
            {
                System.out.println("Please Enter
valid choice ");
            }
```

```
            }
        }
}

}
```

Output:

```
Elements are:
67
89
98
32
Chose one from the below options...

1.Push
```

After poping