

## LAB EXAM DSA

**Name: Vishal Murudkar**

**PRN: 49**

**1. Write a Java program to**

**a. Perform inorder tree traversal**

**Code:**

**Inside main:**

```
package com.inorder.main;
```

```
import com.TreeTraversal.entity.Node;
```

```
import com.TreeTraversal.entity.TreeTraversal;
```

```
public class Inorder {
```

```
    public static void main(String[] args) {
```

```
        TreeTraversal IT = new TreeTraversal();
```

```
        IT.root = new Node(8);
```

```
        IT.root.left = new Node(6);
```

```
        IT.root.right = new Node(10);
```

```
        IT.root.left.left = new Node(5);
```

```
        IT.root.left.right = new Node(7);
```

```
        IT.root.right.left = new Node(9);
```

```
        IT.root.right.right = new Node(11);
```

```
        System.out.println("Inorder traversal: ");
```

```
        IT.printInorder();
```

```
    }
```

```
}
```

**Inside Node:**

```
package com.TreeTraversal.entity;
```

```
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;
```

```
    }
```

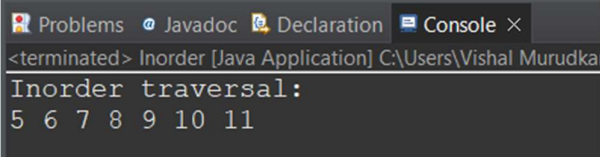
```
}
```

### Inside TreeTraversal:

```
package com.TreeTraversal.entity;
```

```
public class TreeTraversal {  
    public Node root;  
  
    public TreeTraversal() {  
        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);  
    }  
}
```

### Output:



The screenshot shows a Java IDE with a console window open. The console displays the output of an inorder traversal: "Inorder traversal:" followed by the numbers "5 6 7 8 9 10 11" on the next line. The IDE tabs at the top include "Problems", "Javadoc", "Declaration", and "Console".

### b. Implement stack using array

#### Code:

```
package com.stack_array.main;  
import java.util.Scanner;
```

```
public class Stack {  
    int top;  
    int maxsize = 10;  
    int[] arr = new int[maxsize];
```

```
boolean isEmpty()  
{  
    return (top < 0);  
}
```

```
Stack()  
{  
    top = -1;  
}
```

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

```
boolean pop ()  
{  
    if (top == -1)  
    {  
        System.out.println("Underflow !!");  
        System.out.println();  
        return false;  
    }  
    else  
    {  
        top --;  
        System.out.println("Item popped");  
        System.out.println();  
        return true;  
    }  
}
```

```
    }  
}
```

```
void display ()  
{  
    System.out.println("Printing stack elements .....");  
    for(int i = top; i>=0;i--)  
    {  
        System.out.println(arr[i]);  
    }  
    System.out.println();  
}
```

```
public static void main(String[] args) {  
    int choice=0;  
    Scanner sc = new Scanner(System.in);  
    Stack s = new Stack();  
    while(choice != 4)  
    {  
        System.out.println("1- Push\n2- Pop\n3- Show\n4- Exit");  
        System.out.println("Enter your choice");  
        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:

```

<terminated> Stack [Java Application] C:\Users\Vishal Muru
1- Push
2- Pop
3- Show
4- Exit
Enter your choice
1
Enter Value
15
Item pushed

1- Push
2- Pop
3- Show
4- Exit
Enter your choice
1
Enter Value
16
Item pushed

1- Push
2- Pop
3- Show
4- Exit
Enter your choice
1
Enter Value
25
Item pushed

1- Push
2- Pop
3- Show
4- Exit

```

Enter your choice

1

Enter Value

48

Item pushed

1- Push

2- Pop

3- Show

4- Exit

Enter your choice

1

Enter Value

100

Item pushed

1- Push

2- Pop

3- Show

4- Exit

Enter your choice

1

Enter Value

57

Item pushed

1- Push

2- Pop

3- Show

4- Exit

Enter your choice

3

Printing stack elements .....

57

100

48

25

16

15

1- Push

2- Pop

3- Show

4- Exit

Enter your choice

4

Exiting....