

OBJECT ORIENTED PROGRAMMING LAB**Experiment No.: 24****AIM:**

Program to create a generic stack and do the Push and Pop operations.

PROCEDURE:

```
public class Stack {  
    private int arr[];  
    private int top;  
    private int capacity;  
    Stack(int size) {  
        arr = new int[size];  
        capacity = size;  
        top = -1;  
    }  
    public void push(int x) {  
        if (isFull()) {  
            System.out.println("Stack OverFlow");  
            System.exit(1);  
        }  
        System.out.println("Inserting " + x);  
        arr[++top] = x;  
    }  
    public int pop() {  
        if (isEmpty()) {  
            System.out.println("STACK EMPTY");  
        }  
    }  
}
```

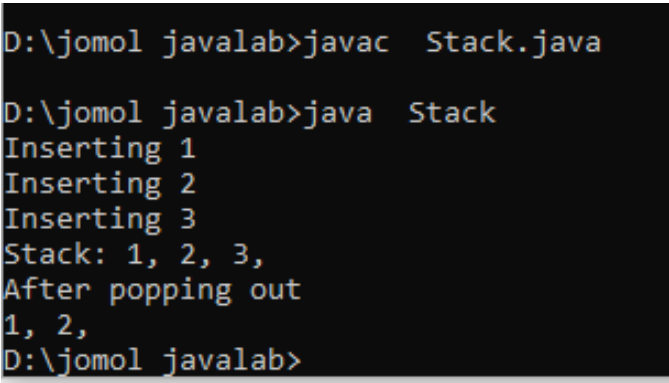
Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

Date: 31/05/2022

```
        System.exit(1);
    }
    return arr[top--];
}
public int getSize() {
    return top + 1;
}
public Boolean isEmpty() {
    return top == -1;
}
public Boolean isFull() {
    return top == capacity - 1;
}
public void printStack() {
    for (int i = 0; i <= top; i++) {
        System.out.print(arr[i] + ", ");
    }
}
public static void main(String[] args) {
    Stack stack = new Stack(5);
    stack.push(1);
    stack.push(2);
    stack.push(3);
    System.out.print("Stack: ");
    stack.printStack();
    stack.pop();
    System.out.println("\nAfter popping out");
    stack.printStack();
}}
```

Output Screenshot:A screenshot of a Windows command prompt window with a black background and white text. The prompt shows the directory 'D:\jomol javalab' and the execution of 'javac Stack.java' followed by 'java Stack'. The output of the program is displayed, showing the insertion of numbers 1, 2, and 3 into a stack, followed by popping them out, resulting in the stack containing 1 and 2.

```
D:\jomol javalab>javac  Stack.java
D:\jomol javalab>java  Stack
Inserting 1
Inserting 2
Inserting 3
Stack: 1, 2, 3,
After popping out
1, 2,
D:\jomol javalab>
```