

OBJECT ORIENTED PROGRAMMING LAB**Experiment No: (CO4) 22****Aim**

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

Procedure

```
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;  
    }  
}
```

Name: Mathew Sebastian

Roll No: 18

Batch: S2 RMCA B

Date: 31-05-2022

```
public int pop() {

    if (isEmpty()) {
        System.out.println("STACK EMPTY");
        System.exit(1);
    }
    return arr[top--];
}

public int getSize() {
    return top + 1;
}

public Boolean isEmpty() {
    return top == -1;
}

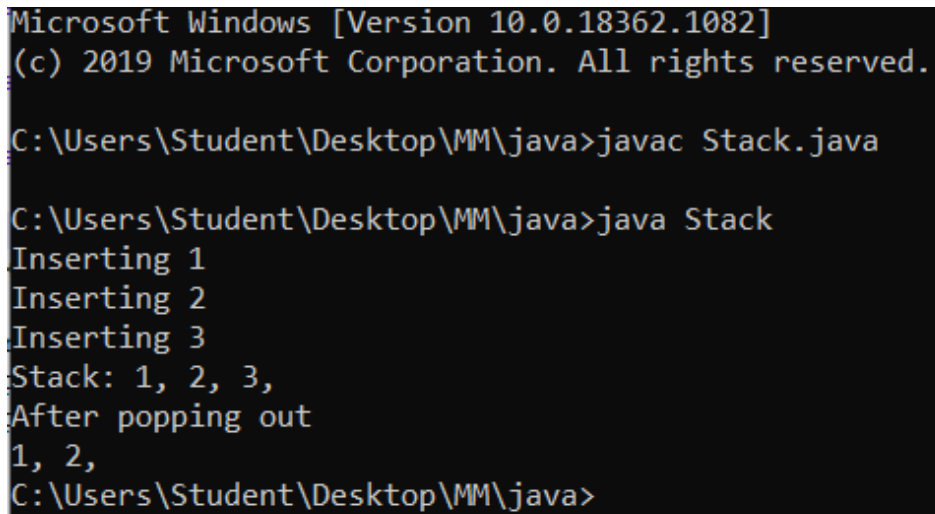
public Boolean isFull() {
    return top == capacity - 1;
}

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

```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Student\Desktop\MM\java>javac Stack.java

C:\Users\Student\Desktop\MM\java>java Stack
Inserting 1
Inserting 2
Inserting 3
Stack: 1, 2, 3,
After popping out
1, 2,
C:\Users\Student\Desktop\MM\java>
```