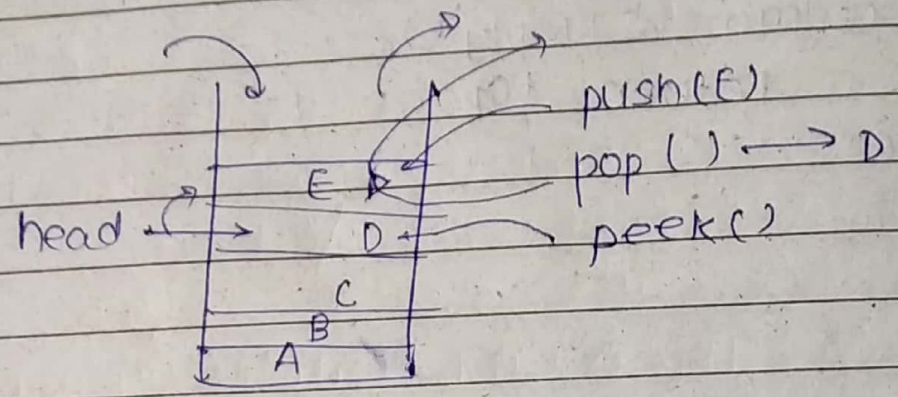
Implementation of stack using Arrays.

```
class MyStack {
    int a[];
    int top;
    int capacity;
    public MyStack (int capacity) {
        this.capacity = capacity;
        top = -1;
        a = new int[capacity];
    }

    public void push (int data) {
        if (top == capacity - 1) {
            System.out.println ("Stack
                overflow");
        }
        top++;
        a[top] = data;
    }

    int
    void pop() {
        if (top == -1) {
            System.out.println ("Stack
                underflow");
        }
        int res = a[top];
        top--;
        return res;
    }
```

```
int peek(){
    if(top == -1){
        System.out.println("stack is empty");
    }
    return a[top];
}

boolean isEmpty(){
    return top == -1;
}
}
```

Methods of stack

→ push (E e) → koi element daalna hai
→ pop() → element bahar nikalna hai
→ peek() → Top ka element
→ search (E e)
→ empty()




```
package vectorAndStacks;
import java.util.Stack;

public class Mainclass {
    public static void main (String[] aregs){

        Stack<Integer> stack = new Stack();

        stack.push (12);
        stack.push (24);
        int popped = stack.pop();
        System.out.println (popped);
        popped = stack.pop();
        System.out.println (popped);
```

```
int peeked = stack.peek();
System.out.println(peeked);
}
}
```

Output:

36
24
24

** Agr push se zyada bar pop ko call krnge i.e. empty stack pr pop ko call krnge then EmptyStackException aa jaega.

Implementation of Stack class

↓

functionalities

↓

push(e)
pop() → e
peek() → e