**Name:** Dev Kamlesh Bhanushali

**PRN:** 22070126032

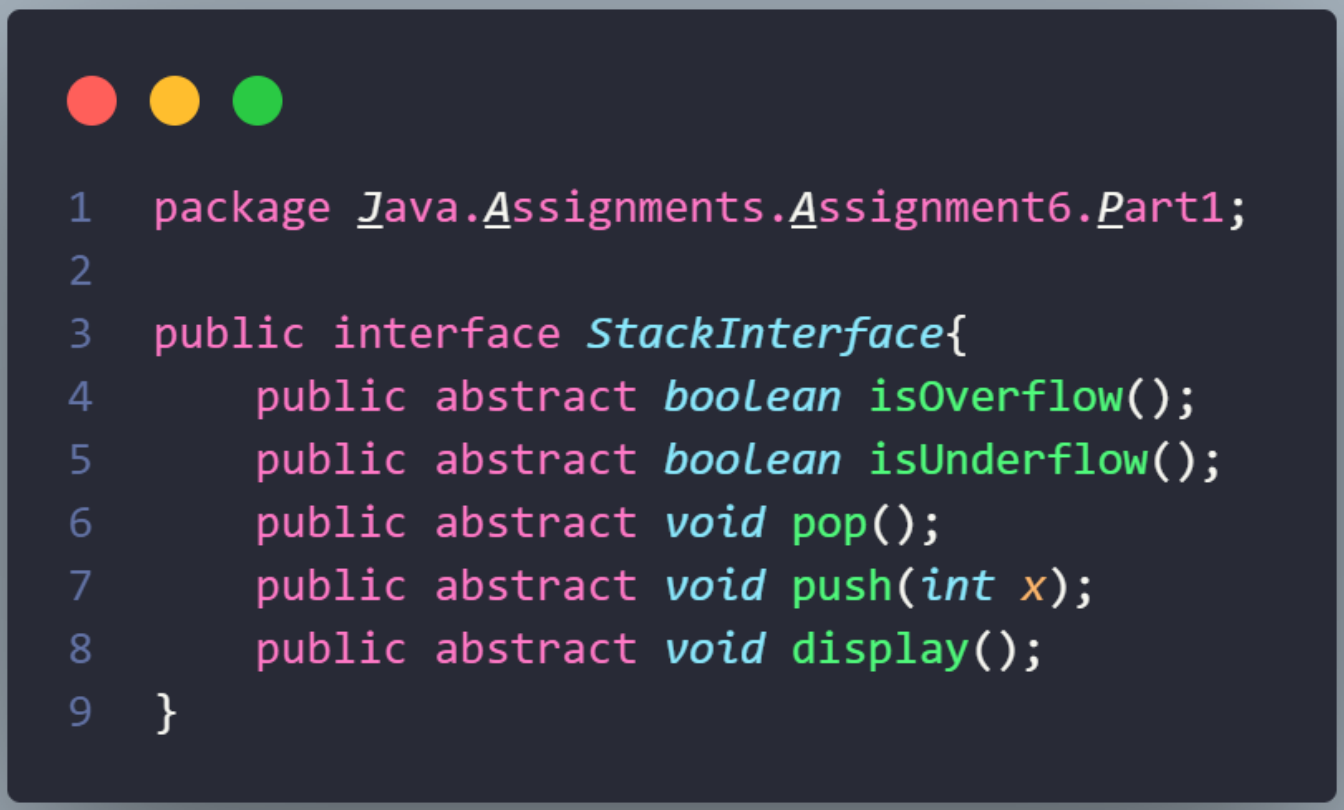
# Java Assignment 6

## Part - 1

Stack Interface

**Github:** https://github.com/devilb2103/Sem-4/tree/main/Java/Assignments/Assignment6

```java
package Java.Assignments.Assignment6.Part1;

public class MainClass {
    public static void main(String[] args) {
        // Fixed f_stack
        System.out.println("Fixed Stack Output ------------------------------");
        FixedStack f_stack = new FixedStack(12);
        f_stack.push(0);
        f_stack.push(2);
        f_stack.push(4);
        f_stack.push(6);
        f_stack.push(8);
        f_stack.display();
        f_stack.pop();
        f_stack.push(10);
        f_stack.pop();
        f_stack.pop();
        f_stack.pop();
        f_stack.display();
        f_stack.pop();
        f_stack.pop();
        f_stack.display();
        System.out.println(f_stack.isOverflow());
        System.out.println(f_stack.isUnderflow());

        try {
            f_stack.pop();
        } catch (RuntimeException e) {
            System.out.println(e.getMessage());
        }

        System.out.println("\n");

        // Dynamic f_stack
        System.out.println("Dynamic Stack Output ------------------------------");
        DynamicStack d_stack = new DynamicStack();
        d_stack.push(0);
        d_stack.push(2);
        d_stack.push(4);
        d_stack.push(6);
        d_stack.push(8);
        d_stack.display();
        d_stack.pop();
        d_stack.push(10);
        d_stack.pop();
        d_stack.pop();
        d_stack.pop();
        d_stack.display();
        d_stack.pop();
        d_stack.pop();
        d_stack.display();
        System.out.println(d_stack.isOverflow());
        System.out.println(d_stack.isUnderflow());

        try {
            d_stack.pop();
        } catch (RuntimeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```



```java
package Java.Assignments.Assignment6.Part1;

public interface StackInterface{
    public abstract boolean isOverflow();
    public abstract boolean isUnderflow();
    public abstract void pop();
    public abstract void push(int x);
    public abstract void display();
}
```

```java
package Java.Assignments.Assignment6.Part1;

public class FixedStack implements StackInterface{

    private int sizeState = 0;
    // sizeState = 0 means normal stack state
    // sizeState = -1 means stack underflow
    // sizeState = 1 means stack overflow
    private int insertPos = 0;
    private int[] arr;

    FixedStack(int size){
        if(size < 1){
            throw new IllegalArgumentException("Fixed Stack must have size greater than 0");
        }

        // initialize array and values
        this.arr = new int[size];
        for (int i = 0; i < this.arr.length; i++) {
            this.arr[i] = Integer.MIN_VALUE;
        }
    }

    @Override
    public boolean isUnderflow(){
        if(sizeState == -1) return true;
        return false;
    }

    @Override
    public boolean isOverflow(){
        if(sizeState == 1) return true;
        return false;
    }

    @Override
    public void pop() {
        if(sizeState > -1){
            this.arr[insertPos - 1] = Integer.MIN_VALUE;
            this.insertPos--;

            if(this.insertPos == 0){
                this.sizeState = -1;
            }
        }
        else{
            throw new RuntimeException("Stack Underflow");
        }
    }

    @Override
    public void push(int value) {
        if(sizeState < 1){
            this.arr[insertPos] = value;
            this.insertPos++;

            if(this.insertPos >= this.arr.length){
                this.sizeState = 1;
            }
        }
        else{
            throw new RuntimeException("Stack Overflow");
        }
    }

    @Override
    public void display() {
        for (int index = 0; index < this.arr.length; index++) {
            if(this.arr[index] != Integer.MIN_VALUE)
            System.out.println(String.format("Element %d: %d", index, this.arr[index]));
        }
        System.out.println("\n");
    }

}

```

```java
package Java.Assignments.Assignment6.Part1;

import java.util.ArrayList;

public class DynamicStack implements StackInterface{

    private ArrayList<Integer> arr = new ArrayList<Integer>();

    @Override
    public boolean isOverflow() {
        return false;
    }

    @Override
    public boolean isUnderflow() {
        if(!(this.arr.size() > 0)){
            return true;
        }
        return false;
    }

    @Override
    public void pop() {
        if(this.arr.size() > 0){
            this.arr.remove(this.arr.size() - 1);
        }
        else{
            throw new RuntimeException("Stack Underflow");
        }
    }

    @Override
    public void push(int value) {
        this.arr.add(value);
    }

    @Override
    public void display() {
        for (int i = 0; i < this.arr.size(); i++) {
            System.out.println(String.format("Element %d: %d", i, this.arr.get(i)));
        }
        System.out.println("\n");
    }


}

```

# Output

```
Fixed Stack Output ------------------------------
Element 0: 0
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8


Element 0: 0
Element 1: 2




false
true
Stack Underflow


Dynamic Stack Output ------------------------------
Element 0: 0
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8


Element 0: 0
Element 1: 2




false
true
Stack Underflow
```

**Github:** https://github.com/devilb2103/Sem-4/tree/main/Java/Assignments/Assignment6