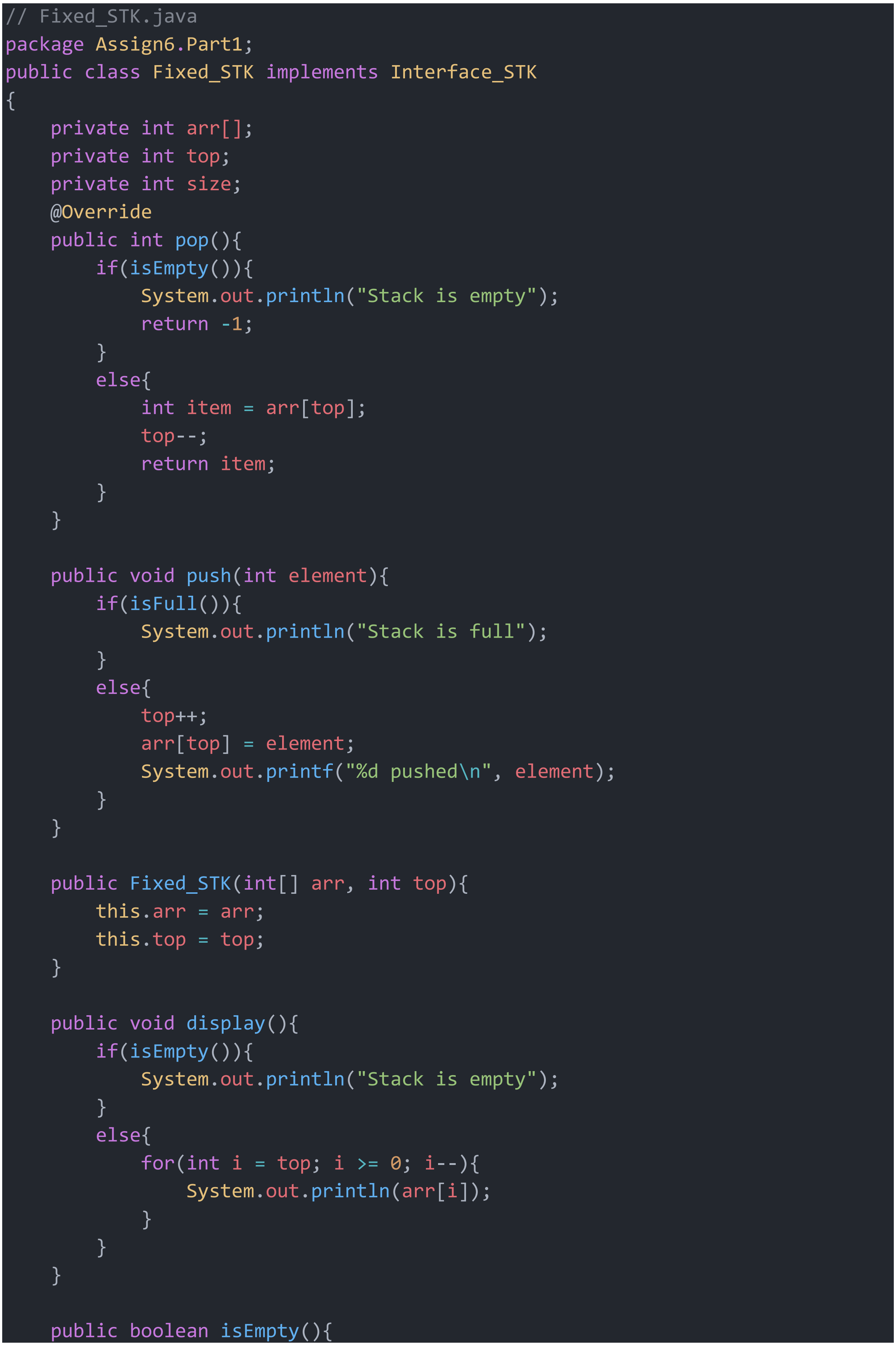
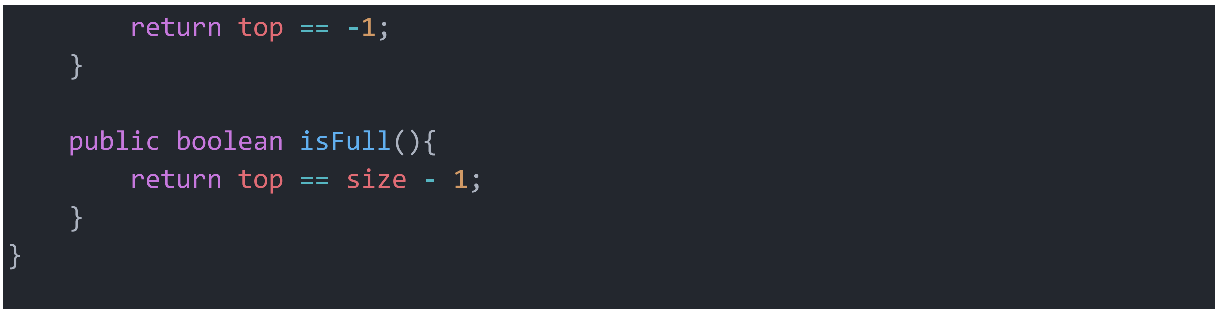
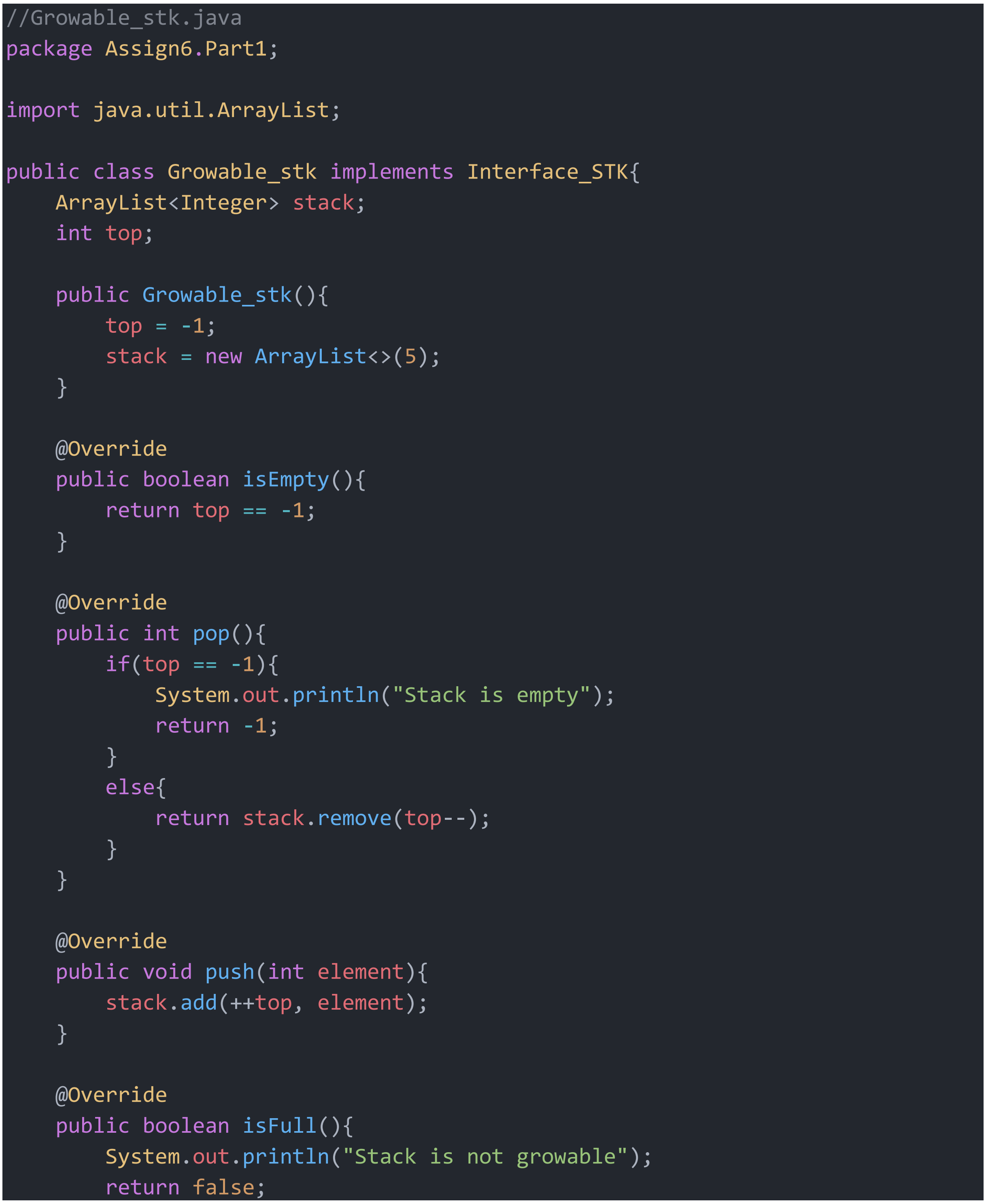
Suyash Tambe Assignment 6



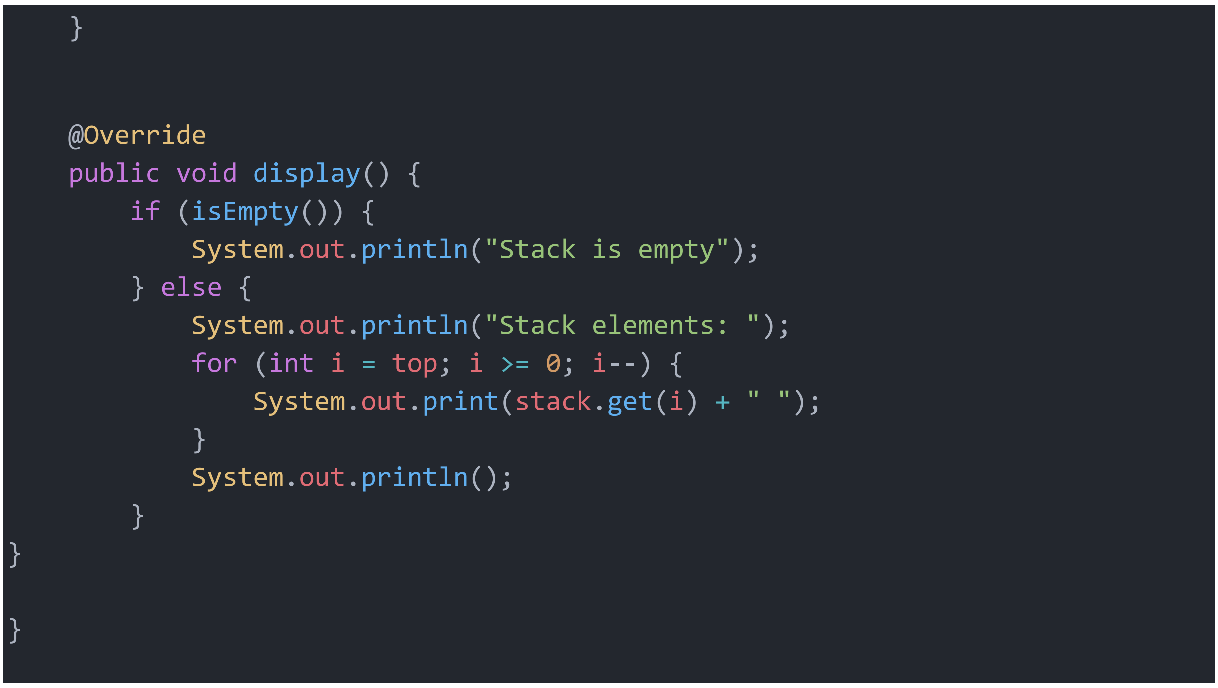
```java
// Fixed_STK.java
package Assign6.Part1;
public class Fixed_STK implements Interface_STK
{
    private int arr[];
    private int top;
    private int size;
    @Override
    public int pop(){
        if(isEmpty()){
            System.out.println("Stack is empty");
            return -1;
        }
        else{
            int item = arr[top];
            top--;
            return item;
        }
    }

    public void push(int element){
        if(isFull()){
            System.out.println("Stack is full");
        }
        else{
            top++;
            arr[top] = element;
            System.out.printf("%d pushed\n", element);
        }
    }

    public Fixed_STK(int[] arr, int top){
        this.arr = arr;
        this.top = top;
    }

    public void display(){
        if(isEmpty()){
            System.out.println("Stack is empty");
        }
        else{
            for(int i = top; i >= 0; i--){
                System.out.println(arr[i]);
            }
        }
    }

    public boolean isEmpty(){
```



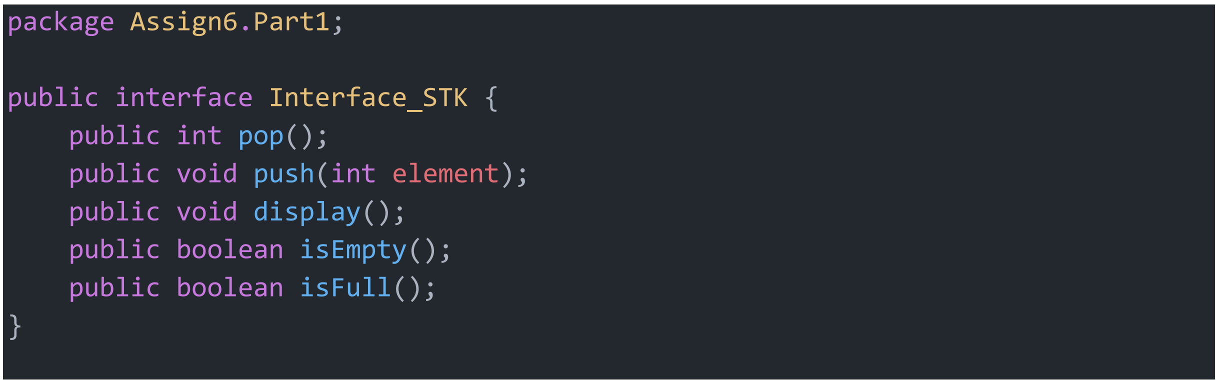
```
        return top == -1;
    }

    public boolean isFull(){
        return top == size - 1;
    }
}
```



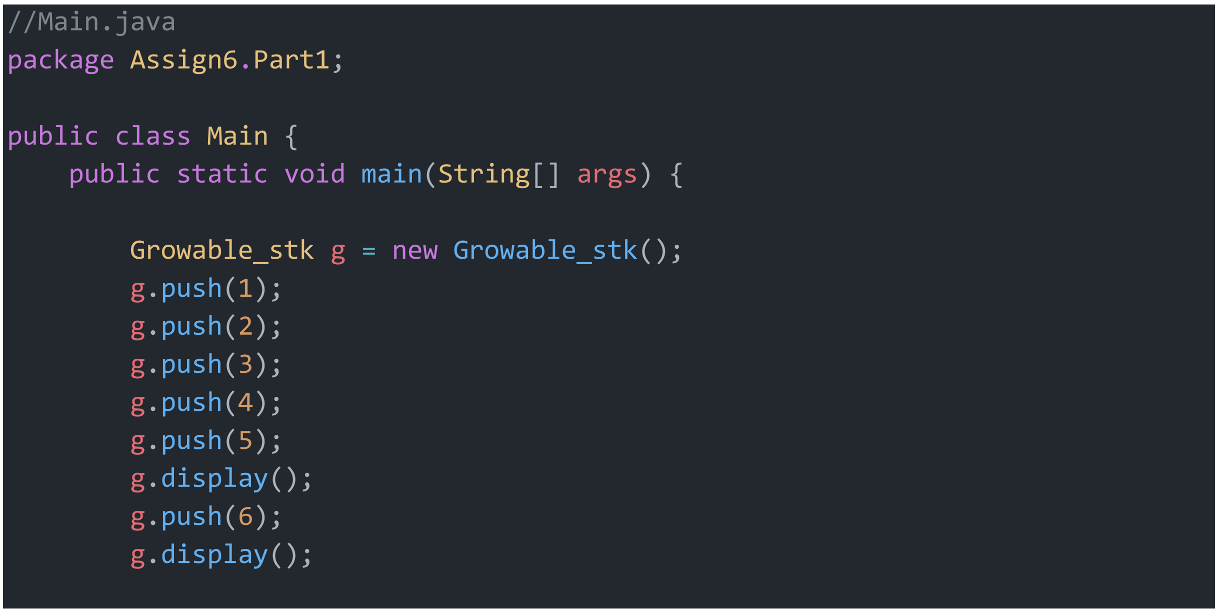
```
//Growable_stk.java
package Assign6.Part1;

import java.util.ArrayList;

public class Growable_stk implements Interface_STK{
    ArrayList<Integer> stack;
    int top;

    public Growable_stk(){
        top = -1;
        stack = new ArrayList<>(5);
    }

    @Override
    public boolean isEmpty(){
        return top == -1;
    }

    @Override
    public int pop(){
        if(top == -1){
            System.out.println("Stack is empty");
            return -1;
        }
        else{
            return stack.remove(top--);
        }
    }

    @Override
    public void push(int element){
        stack.add(++top, element);
    }

    @Override
    public boolean isFull(){
        System.out.println("Stack is not growable");
        return false;
```



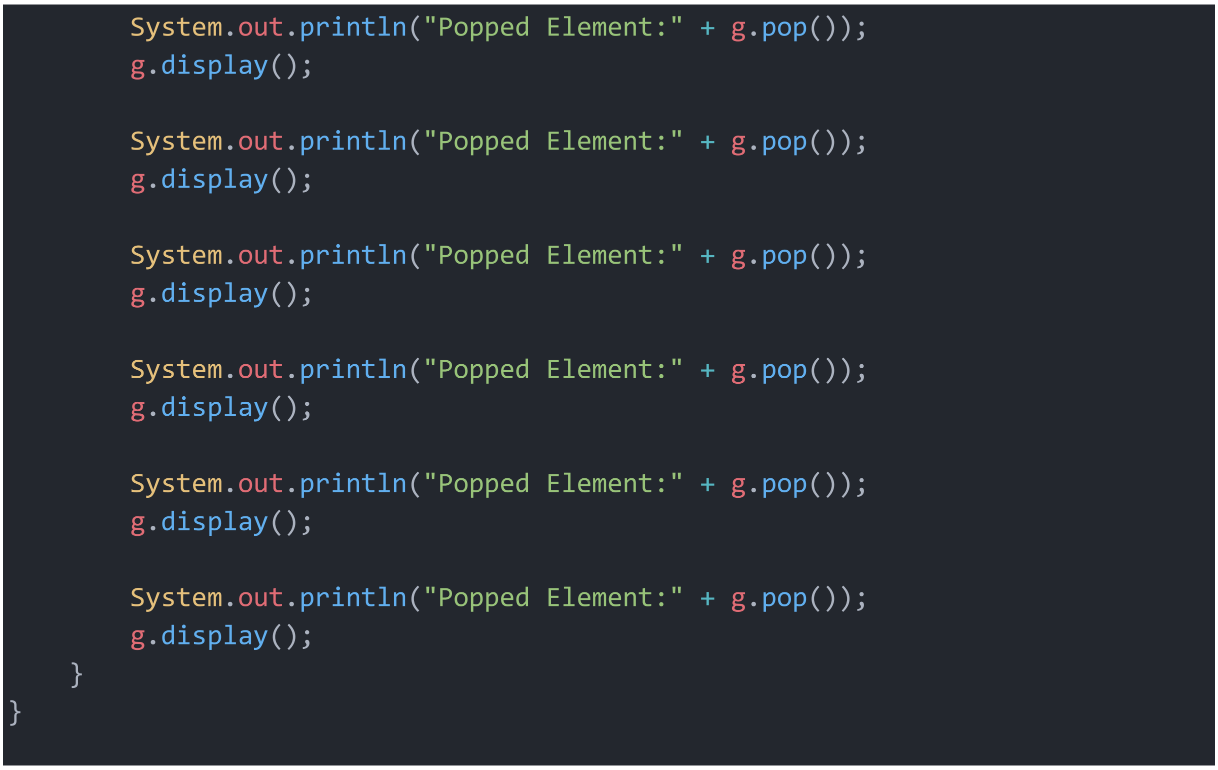
```java
    }


    @Override
    public void display() {
        if (isEmpty()) {
            System.out.println("Stack is empty");
        } else {
            System.out.println("Stack elements: ");
            for (int i = top; i >= 0; i--) {
                System.out.print(stack.get(i) + " ");
            }
            System.out.println();
        }
}

}
```



```java
package Assign6.Part1;

public interface Interface_STK {
    public int pop();
    public void push(int element);
    public void display();
    public boolean isEmpty();
    public boolean isFull();
}
```



```java
//Main.java
package Assign6.Part1;

public class Main {
    public static void main(String[] args) {

        Growable_stk g = new Growable_stk();
        g.push(1);
        g.push(2);
        g.push(3);
        g.push(4);
        g.push(5);
        g.display();
        g.push(6);
        g.display();
```



```
        System.out.println("Popped Element:" + g.pop());
        g.display();

        System.out.println("Popped Element:" + g.pop());
        g.display();

        System.out.println("Popped Element:" + g.pop());
        g.display();

        System.out.println("Popped Element:" + g.pop());
        g.display();

        System.out.println("Popped Element:" + g.pop());
        g.display();

        System.out.println("Popped Element:" + g.pop());
        g.display();
    }
}
```