Assignment-2

Implement a stack data structure in Java. A stack is a linear data structure that follows the Last In First Out (LIFO) principle. Your task is to create a class Stack that supports the following operations:

push(element): Add an element to the top of the stack.
pop(): Remove and return the top element of the stack. If the stack is empty, return an appropriate error message or exception.
peek(): Return the top element of the stack without removing it. If the stack is empty, return an appropriate error message or exception.
isEmpty(): Return true if the stack is empty, otherwise return false.
size(): Return the number of elements in the stack.

```
package datastructures.linear;


class StackCreation{
	static final int MAX=1000;
	int top;
	int a[]=new int[MAX];

	boolean isEmpty() {
		return (top<0);
	}
```

```
StackCreation(){
        top=-1;
}

boolean push(int x) {
        if(top>=(MAX-1)) {
                System.out.println("Stack Overflow");
                return false;
        }
        else {
                a[++top]=x;
                System.out.println(x+" pushed ints stack");
                return true;
        }
}
int pop() {
        if(isEmpty()) {
                System.out.println("stack underflow");
                return 0;
        }
        else {
                int ele;
                ele=a[top];
                --top;
                return ele;
        }
```

```
	}

	void print() {
		for(int i=top;i>-1;i--) {
			System.out.println(" "+a[i]);
		}
	}

	int peek() {
		if(top<0) {
			System.out.println("Stack underflow");
			return 0;
		}
		else {
			int x=a[top];
			return x;
		}
	}



}
public class stack {

	public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
        StackCreation sc=new StackCreation();
        sc.push(10);
        sc.push(20);
        sc.push(30);
        sc.push(40);
        sc.push(50);
        System.out.println("Stack elements are: ");
        sc.print();
        System.out.println("------------------------");
        System.out.println(sc.pop()+"popped  from stack");
        sc.print();
        System.out.println("top element is :"+sc.peek());

    }

}
```