```
class Stack {
	private int top; // represents the index position of the top most element in the stack
	private int maxSize; // represents the maximum number of elements that can be stored in the stack
	private int[] arr;

	Stack(int maxSize) {
		this.top = -1; // top is -1 when the stack is created
		this.maxSize = maxSize;
		arr = new int[maxSize];
	}

	// Checking if the stack is full or not
	public boolean isFull() {
		if (top >= (maxSize - 1)) {
			return true;
		}
		return false;
	}

	// Adding a new element to the top of the stack
	public boolean push(int data) {
		if (isFull()) {
			return false;
		} else {
			arr[++top] = data;
			return true;
		}
	}

	// Returning the top most element of the stack
```

```
	public int peek() {
		if (isEmpty())
			return Integer.MIN_VALUE;
		else
			return arr[top];
	}

	// Displaying all the elements of the stack
	public void display() {
		if (isEmpty())
			System.out.println("Stack is empty!");
		else {
			System.out.println("Displaying stack elements");
			for (int index = top; index >= 0; index--) {
				System.out.println(arr[index]); // accessing element at position
index
			}
		}
	}

	// Checking if the stack is empty or not
	public boolean isEmpty() {
		if (top < 0) {
			return true;
		}
		return false;
	}

	// Removing the element from the top of the stack
	public int pop() {
		if (isEmpty())
```

```
                return Integer.MIN_VALUE;
        else
                return arr[top--];
    }
}

class Tester {
    public static void main(String args[]) {
        Stack stack = new Stack(5);
        System.out.println("Stack created.\n");

        if (stack.push(1))
            System.out.println("The element is pushed to the stack!\n");
        else
            System.out.println("Stack is full!\n");

        if (stack.push(2))
            System.out.println("The element is pushed to the stack!\n");
        else
            System.out.println("Stack is full!\n");

        if (stack.push(3))
            System.out.println("The element is pushed to the stack!\n");
        else
            System.out.println("Stack is full!\n");

        if (stack.push(4))
            System.out.println("The element is pushed to the stack!\n");
        else
            System.out.println("Stack is full!\n");
```

```
if (stack.push(5))
        System.out.println("The element is pushed to the stack!\n");
else
        System.out.println("Stack is full!\n");

stack.display();

if (stack.push(6))
        System.out.println("The element is pushed to the stack!\n");
else
        System.out.println("Stack is full!\n");

System.out.println("The top element is : " + stack.peek());

int poppedElement = stack.pop();
if (poppedElement == Integer.MIN_VALUE)
        System.out.println("Stack is empty\n");
else
        System.out.println("The element popped out is : " + poppedElement + "\n");

poppedElement = stack.pop();
if (poppedElement == Integer.MIN_VALUE)
        System.out.println("Stack is empty\n");
else
        System.out.println("The element popped out is : " + poppedElement + "\n");

poppedElement = stack.pop();
if (poppedElement == Integer.MIN_VALUE)
        System.out.println("Stack is empty\n");
else
        System.out.println("The element popped out is : " + poppedElement + "\n");
```

```
                poppedElement = stack.pop();
                if (poppedElement == Integer.MIN_VALUE)
                        System.out.println("Stack is empty\n");
                else
                        System.out.println("The element popped out is : " + poppedElement + "\n");

                poppedElement = stack.pop();
                if (poppedElement == Integer.MIN_VALUE)
                        System.out.println("Stack is empty\n");
                else
                        System.out.println("The element popped out is : " + poppedElement + "\n");

                poppedElement = stack.pop();
                if (poppedElement == Integer.MIN_VALUE)
                        System.out.println("Stack is empty\n");
                else
                        System.out.println("The element popped out is : " + poppedElement + "\n");
        }
}
```

Output:

Stack created.

The element is pushed to the stack!

The element is pushed to the stack!

The element is pushed to the stack!

The element is pushed to the stack!

The element is pushed to the stack!

Displaying stack elements

5

4

3

2

1

Stack is full!

The top element is : 5

The element popped out is : 5

The element popped out is : 4

The element popped out is : 3

The element popped out is : 2

The element popped out is : 1

Stack is empty