

```

//Saket M Kharche
// Stack implementation using an array
class StackArray {
    private int[] stackArray; // Array to hold stack elements
    private int top; // Index of the top element in the stack

    // Constructor to initialize the stack with a specified size
    public StackArray(int size) {
        stackArray = new int[size]; // Initialize the array with the given size
        top = -1; // Set top to -1 to indicate the stack is empty
    }

    // Check if the stack is empty
    public boolean isEmpty() {
        return top == -1; // Stack is empty if top is -1
    }

    // Check if the stack is full
    public boolean isFull() {
        return top == stackArray.length - 1; // Stack is full if top is at the last index
    }

    // Push an element onto the stack
    public void push(int value) {
        if (isFull()) { // If the stack is full, we cannot push
            System.out.println("Stack is full! Cannot push " + value);
        } else {
            stackArray[++top] = value; // Increment top and insert the value at the top
            System.out.println("Pushed " + value + " to the stack.");
        }
    }

    // Pop an element from the stack
    public int pop() {
        if (isEmpty()) { // If the stack is empty, return -1 indicating failure
            System.out.println("Stack is empty! Cannot pop.");
            return -1;
        } else {
            return stackArray[top--]; // Return the top element and decrease the top index
        }
    }

    // Peek at the top element of the stack without removing it
    public int peek() {
        if (isEmpty()) { // If the stack is empty, return -1
            System.out.println("Stack is empty! Cannot peek.");
            return -1;
        } else {
            return stackArray[top]; // Return the top element
        }
    }
}

public class MainArray {
    public static void main(String[] args) {
        StackArray stack = new StackArray(5); // Create a stack with size 5

        stack.push(10);
        stack.push(20);
        stack.push(30);
        stack.push(40);
        stack.push(50);
        stack.push(60); // This will show that the stack is full

        System.out.println("Top element is: " + stack.peek()); // Peek at the top element

        System.out.println("Popped element: " + stack.pop()); // Pop the top element
        System.out.println("Popped element: " + stack.pop()); // Pop another element
    }
}

```