public class Stack {

private int maxSize;

private int[] stackArray;

private int top;

// Constructor

public Stack(int size) {

maxSize = size;

stackArray = new int[maxSize];

top = -1;

}

// Push element onto stack

public void push(int value) {

if (isFull()) {

System.out.println("Stack is full. Cannot push " + value);

return;

}

stackArray[++top] = value;

}

// Pop element from stack

public int pop() {

if (isEmpty()) {

System.out.println("Stack is empty. Cannot pop.");

return -1; // or throw exception

}

return stackArray[top--];

}

// Peek top element without removing

public int peek() {

if (isEmpty()) {

System.out.println("Stack is empty.");

return -1; // or throw exception

}

return stackArray[top];

}

// Check if stack is empty

public boolean isEmpty() {

return (top == -1);

}

// Check if stack is full

public boolean isFull() {

return (top == maxSize - 1);

}

// Get current size of stack

public int size() {

return top + 1;

}

public static void main(String[] args) {

Stack stack = new Stack(5);

stack.push(10);

stack.push(20);

stack.push(30);

System.out.println("Top element: " + stack.peek());

while (!stack.isEmpty()) {

System.out.println("Popped: " + stack.pop());

}

}

}