**COMP 203 Lab 5 Solution**

**Stack**

1. import java.util.EmptyStackException;

public class ArrayStack {

private int[] array;

private int size;

private static final int DEFAULT\_CAPACITY = 10;

public ArrayStack() {

array = new int[DEFAULT\_CAPACITY];

size = 0;

}

public void push(int element) {

if (size == array.length) {

throw new IllegalStateException("Stack is full");

}

array[size++] = element;

}

public int pop() {

if (isEmpty()) {

throw new EmptyStackException();

}

return array[--size];

}

public int size() {

return size;

}

public boolean isEmpty() {

return size == 0;

}

public static void main(String[] args) {

// Testing the integer-specific ArrayStack

ArrayStack stack = new ArrayStack();

// Pushing elements onto the stack

stack.push(10);

stack.push(20);

stack.push(30);

// Printing the size of the stack

System.out.println("Size of the stack: " + stack.size());

// Popping elements from the stack and printing them

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

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

// Printing the size of the stack after popping elements

System.out.println("Size of the stack: " + stack.size());

}

}

2. class Node {

String data;

Node next;

public Node(String data) {

this.data = data;

this.next = null;

}

}

public class SLLStack {

private Node head;

private int size;

public SLLStack() {

this.head = null;

this.size = 0;

}

public void push(String data) {

Node newNode = new Node(data);

newNode.next = head;

head = newNode;

size++;

}

public String pop() {

if (isEmpty()) {

return null;

}

String poppedData = head.data;

head = head.next;

size--;

return poppedData;

}

public int size() {

return size;

}

public boolean isEmpty() {

return size == 0;

}

public static void main(String[] args) {

// Testing the String SLLStack

SLLStack stack = new SLLStack();

// Pushing elements onto the stack

stack.push("One");

stack.push("Two");

stack.push("Three");

// Printing the size of the stack

System.out.println("Size of the stack: " + stack.size());

// Popping elements from the stack and printing them

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

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

// Printing the size of the stack after popping elements

System.out.println("Size of the stack: " + stack.size());

}

}