|  |  |
| --- | --- |
| Student Name | Asadullah |
| Roll Number | 21SW036 |
| Section # | 03 |
| Lab # | 09 |

**Task#01**

Question statement

Question#01: To develop stack using Array (using generic class)

a)Implement the isEmpty() method in the Stack

b) Implement the getSize() method in the Stack

c) Implement the search() method in the Stack to check whether the element exists in the stack or not.

d) Implement the push() method in the Stack and then Push 10 values.

e) Implement the pop() method in the Stack and print the stack using pop.

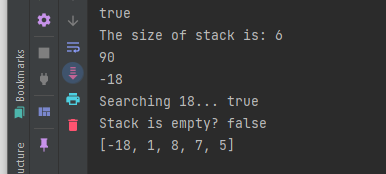
f) Display the all stack data using display() method..

# Q1.Java

**Code:**

import java.util.EmptyStackException;  
  
public class Q1 {  
  
 static class ArrayStack<k>{  
 private int size = 0;  
 private k [] arr;  
  
 public ArrayStack(int capacity){  
 arr = (k[]) new Object[capacity];  
 } // end of constructor  
  
 public void push(k stack){  
 if(size==arr.length){  
 resize();  
 }  
 arr[size++] = stack;  
 } // end of push() method  
  
 public k pop() {  
 if (size==0){  
 throw new EmptyStackException();  
 }  
 k obj = arr[--size];  
 return obj;  
 } // end of pop() method  
 public boolean search(k obj){  
 for (int i=0; i< arr.length; i++){  
 if (arr[i]==obj){  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public k peek() {  
 if(size==0){  
 throw new EmptyStackException();  
 }  
 return arr[size-1];  
 } // end of peek() method  
 public void display(){  
 System.*out*.print("[");;  
 while (!isEmpty()){  
 if (getSize()==1)  
 System.*out*.print(peek());  
 else  
 System.*out*.print(peek()+", ");  
 pop();  
 }  
 System.*out*.println("]");  
 } // end of display()  
 public boolean isEmpty(){  
 return (size==0);  
 }  
 public int getSize(){  
 return size;  
 }  
 private void resize(){  
 Object [] aa = arr;  
 arr = (k[]) new Object[2\*aa.length];  
 System.*arraycopy*(aa, 0, arr, 0, size);  
 } // end of resize() method  
 } // end of ArrayStack class  
  
 public static void main(String[] args) {  
  
 ArrayStack<Integer> arrayStack = new ArrayStack<>(10);  
  
 System.*out*.println(arrayStack.isEmpty());  
  
 arrayStack.push(5);  
 arrayStack.push(7);  
 arrayStack.push(8);  
 arrayStack.push(1);  
 arrayStack.push(-18);  
 arrayStack.push(90);  
  
 System.*out*.println("The size of stack is: "+arrayStack.getSize());  
 System.*out*.println(arrayStack.peek());  
  
 arrayStack.pop();  
 System.*out*.println(arrayStack.peek());  
  
 System.*out*.println("Searching 18... "+arrayStack.search(-18));  
  
 System.*out*.println("Stack is empty? "+arrayStack.isEmpty());  
 arrayStack.display();  
  
 }  
}

**Output:**

****

**Task#02**

Question#02: To develop stack using Linked List implemented in the previous lab (using generic class)

a)Implement the isEmpty() method in the Stack Happy Coding   
b) Implement the getSize() method in the Stack

c) Implement the search() method in the Stack to check whether the element exists in the stack or not.

d) Implement the push() method in the Stack and then Push 10 values.

e) Implement the pop() method in the Stack and print the stack using pop.

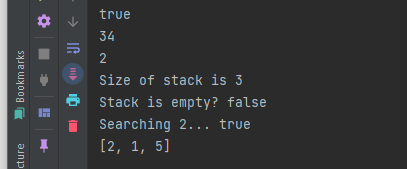
f) Display the all stack data using display() method..

# Q2.Java

**Code:**

public class Q2 {  
  
 static class LinkedStack<k> {  
 static class Node<k>{  
 Node<k> next;  
 k val;  
  
 public Node(k val) {  
 this.val = val;  
 }  
 } // end of class Node  
  
 private Node<k> top;  
 private int length = 0;  
  
 public LinkedStack(){  
 top = null;  
 length = 0;  
 }  
 public void push(k data){  
 Node<k> temp = new Node<k>(data);  
 temp.next = top;  
 top = temp;  
 length++;  
 }  
 public k pop(){  
 k result = top.val;  
 top = top.next;  
 length--;  
 return result;  
 }  
 public k peek(){  
 return top.val;  
 }  
 public int getSize(){ return length; }  
 public void display() {  
 System.*out*.print("[");  
 ;  
 while (!isEmpty()) {  
 if (getSize() == 1)  
 System.*out*.print(peek());  
 else  
 System.*out*.print(peek() + ", ");  
 pop();  
 }  
 System.*out*.println("]");  
 }  
 public boolean isEmpty(){  
 return (length == 0);  
 }  
 public boolean search(k val){  
 for (Node p = top; p!=null; p=p.next){  
 if (p.val==val)  
 return true;  
 }  
 return false;  
 }  
 } // end of class LinkedStack  
  
  
 public static void main(String[] args) {  
  
 LinkedStack<Integer> linkedStack = new LinkedStack<>();  
  
 System.*out*.println(linkedStack.isEmpty());  
  
 linkedStack.push(5);  
 linkedStack.push(1);  
 linkedStack.push(2);  
 linkedStack.push(34);  
  
 System.*out*.println(linkedStack.peek());  
  
 linkedStack.pop();  
 System.*out*.println(linkedStack.peek());  
  
 System.*out*.println("Stack is empty? "+linkedStack.isEmpty());  
 System.*out*.println("Searching 2... "+linkedStack.search(2));  
  
 linkedStack.display();  
  
 } // end of main()'  
}

**Output:**

****