**Report On**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Advanced Object Oriented Programming Laboratory**

**(15UCSL605)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Submitted**

**By**

**Name: SHRIKRISHNA BHAT USN: 2SD15CS099**

**VI Semester, B Division Academic Year: 2017-18**

**Belonging to**

**Batch: B2**

**Faculty In-charge: (Faculty in-charge for lab)**

**Course Teacher: Indira R Umarji**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Department of Computer Science & Engineering**

**SDM College of Engineering and Technology**

**Date: 12/02/2018**

**Complete problem definition of the Term work-1**

**Using** state diagram, **design** and **describe** the behavior of STACK which contains maximum of FOUR integer elements.

Implement the above design in JAVA Programming Language. Design the TEST-DRIVER class to include minimum number of TEST CASES to test the complete features of STACK class designed.

**Expected Learning:** How to define the class, Use of Instance Variables, data types, operators, control structures, Understanding of access specifies, Declaring methods, parameterized methods, constructor, Interface, finalize() method, Compilation procedures, use of package, class path and other basic features.

**Source code/Program**

**Contents of package stackpack**

/\*\* Java program for test-driver Stack

Author: Shrikrishna Bhat

USN: 2SD15CS099

\*/

package stack;// to package stack

import java.util.\*;//import util.\*

/\*\*

\* Stack class(generic type)

\*/

class Stack<T>{ //Generic Stack

private int size;

private T[] arr;

private int top; // top of stack

/\*\*

\* Constructor for initializing Array.

\*/

@SuppressWarnings("unchecked")

public Stack(int size){

this.size = size;

arr = (T[])new Object[size]; //Creation of Generic Stack Array

top = -1; // initialize Stack to with -1

}

public void display(){

System.out.println("Stack = ");

if (top == -1){

System.out.println("Empty");

return ;

}

for (int i = top; i >= 0; i--)

System.out.print(arr[i]+" ");

System.out.println();

}

/\*\*

\* Push items in stack, it will put items on top of Stack.

\*/

public void push(T value){

if(isFull()){

throw new IndexOutOfBoundsException("Overflow Exception");

}

arr[++top] = value;

}

/\*\*

\* Pop items in stack, it will remove items from top of Stack.

\*/

public T pop(){

if(isEmpty()){

throw new NoSuchElementException("Underflow Exception");

}

return arr[top--]; // remove item and decrement top as well.

}

public T peek(){

if(isEmpty()){

throw new NoSuchElementException("Underflow Exception");

}

return arr[top]; // remove item and decrement top as well.

}

/\*\*

\* @return true if Stack is empty

\*/

public boolean isEmpty(){

return (top == -1);

}

/\*\*

\* @return true if stack is full

\*/

public boolean isFull(){

return (top == size - 1);

}

}

**Contents of main program**

/\*\* Java program for test-driver Stack

Author: Shrikrishna Bhat

USN:2SD15CS099

\*/

import stack.\*;// import package

import java.util.\*;//Import utility class

public class TestStack10{

public static void main(String[] args){

int i;

System.out.println("Stack Test");

System.out.println("-------------------------------------------------------------");

/\* Creating object of class arrayStack \*/

System.out.println("Stack of size 3 created");

Stack<Integer> stk = new Stack<Integer>(10); // Creation of Generic Stack

for(i=1;i<=10;i++){ // Test-Driver

switch (i){

case 1: //Empty stack display

System.out.println("Test case: 1");

stk.display();

System.out.println("Test case 1: \"Empty Stack display\" Passed\n");

break;

case 2: //Empty stack Pop

System.out.println("Test case: 2");

try{

System.out.println("Popped Element = " + stk.pop());

}catch(Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 2: \"Empty Stack Popped\" Passed\n");

break;

case 3: //Push first element i.e '1'

System.out.println("Test case: 3");

try{

stk.push(1);

System.out.println("Peeked element is " +stk.peek());

}catch(Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 3 \"1-PUSH\" passed\n");

break;

case 4: //Push second element i.e '2'

System.out.println("Test case: 4");

try{

stk.push(2);

System.out.println("Peeked element is " +stk.peek());

}catch(Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 4 \"2-PUSH'passed\"\n");

break;

case 5: //Display Partially filled stack

System.out.println("Test case: 5");

stk.display();

System.out.println("Test case 5: \"Partial Full Stack display\" Passed\n");

break;

case 6: //Pop from partially filed stack

System.out.println("Test case: 6");

try{

System.out.println("Popped Element = " + stk.pop());

}catch (Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 6: \"Popped Partially full Stack\" Passed\n");

break;

case 7: //Push three elements to stack i.e '2' & '3'

System.out.println("Test case: 7");

try{

stk.push(2);

stk.push(3);

System.out.println("Peeked element is " +stk.peek());

}catch (Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 7 \"2,3-PUSH'passed\"\n");

break;

case 8: //Display Full Stack

System.out.println("Test case: 8");

stk.display();

System.out.println("Test case 8: \"Full Stack display\" Passed\n");

break;

case 9: //Try to push after stack is full

System.out.println("Test case: 9");

try{

stk.push(4);

}catch (Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 9 \"OverFlow Condition\" passed\n");

break;

case 10: // Pop all elements from stack

System.out.println("Test case: 10");

try{

System.out.println("Popped Element = " + stk.pop());

System.out.println("Popped Element = " + stk.pop());

System.out.println("Popped Element = " + stk.pop());

} catch (Exception e){

System.out.println("Error : " + e.getMessage());

}

System.out.println("Test Case 10 : \"Popped all elements\" Passed\n");

break;

default:

break;

}

}

}

}

**OUTPUT:**

