1. **Stack Operations using Interface: Create an interface Stack with a variable size and abstract methods push(), pop(), display(), overflow(), and underflow(). Implement a subclass IntegerStack that implements the Stack interface. Create a test class to check the working of all methods in the IntegerStack class.**

**Answer:**

interface Stack{

int size = 5;

void push(int element);

int pop();

void display();

boolean overflow();

boolean underflow();

}

class integerStack implements Stack

{

int[] stackArray = new int[size];

int top = -1;

public void push(int element){

if(overflow()){

System.out.println("stack overflow can not push"+element);

}

else{

stackArray[++top]=element;

System.out.println("pushed"+element);

}

}

public int pop(){

if(underflow()){

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

return -1;

}

else{

System.out.println("Popped: " + stackArray[top]);

return stackArray[top--];

}

}

public boolean overflow(){

return top == size -1;

}

public boolean underflow(){

return top == -1;

}

public void display(){

if(underflow()){

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

}

else{

System.out.println("Stack elements is:");

for(int i=0; i<=top; i++){

System.out.println(stackArray[i]+ " ");

}

System.out.println();

}

}

}

class stackIn {

public static void main(String[] args){

integerStack test = new integerStack();

test.pop();

test.display();

test.push(56);

test.push(27);

test.push(17);

test.push(1);

test.push(34);

test.display();

test.push(67);

test.display();

test.pop();

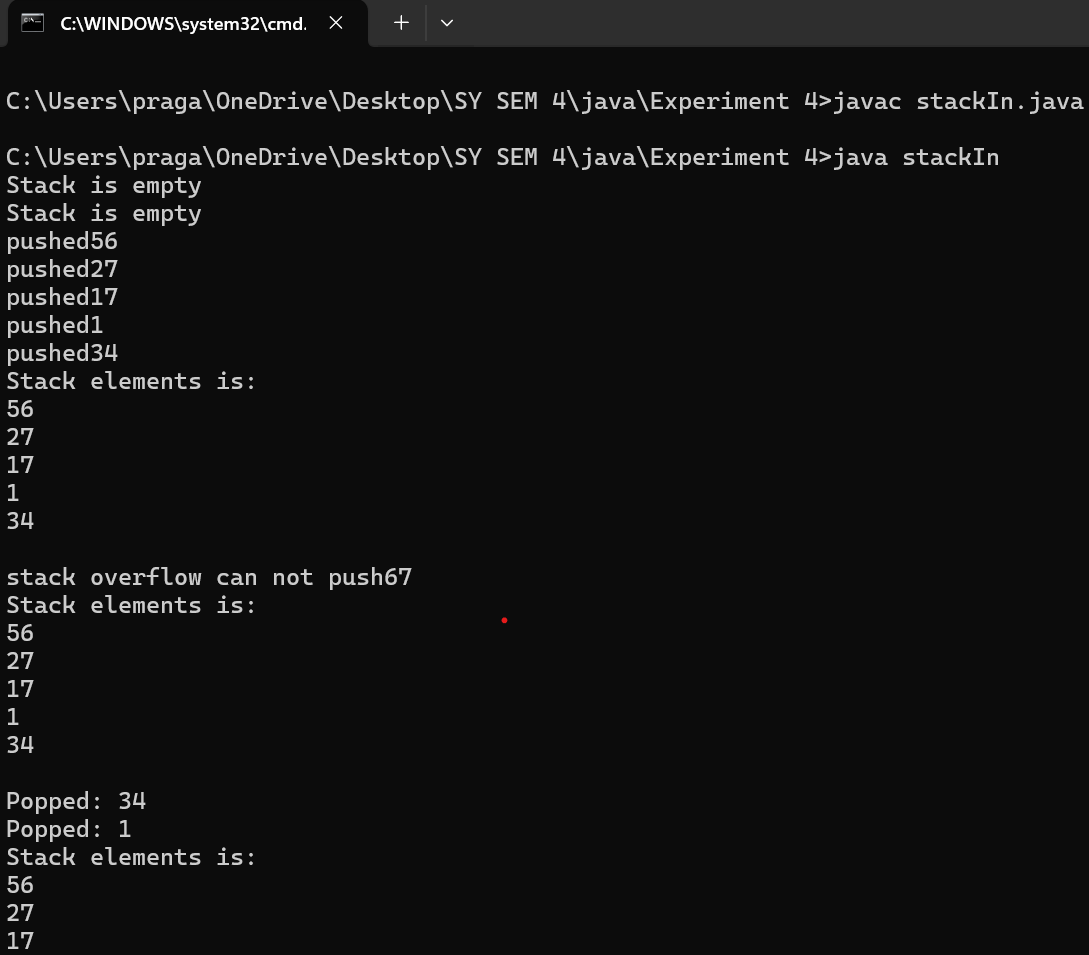
test.pop();

test.display();

}

}

Output:



1. **Shape Interface with Rectangle and Triangle: Implement the following: a. Create an interface Shape with an abstract method area(). b. Create two classes, Rectangle and Triangle, that implement the Shape interface. c. Calculate and display the area of both Rectangle and Triangle.**

**Answer:**

interface shape{

public static final double pi = 3.14;

double complete(double d1,double d2);

double compl(double base, double height);

}

class rectangle implements shape{

public double complete(double d1, double d2){

return d1\*d2;

}

public double compl(double base, double height){

return 0;

}

}

class triangle implements shape {

public double compl(double base, double height){

return 0.5\*base\*height;

}

public double complete(double d1, double d2){

return 0;

}

}

public class shapeDemo {

public static void main(String[] args) {

rectangle r1 = new rectangle();

triangle r2 = new triangle();

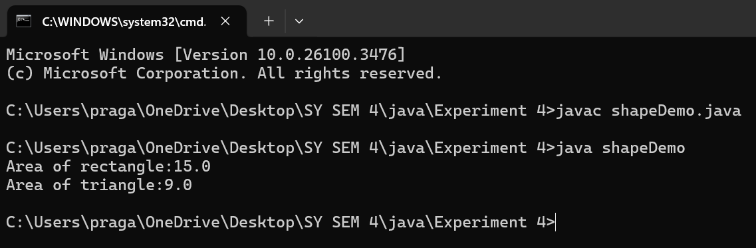
System.out.println("Area of rectangle:"+ r1.complete(3,5));

System.out.println("Area of triangle:"+ r2.compl(3,6));

}

}

**Output:**

****

1. **Student Exam Results Using Inheritance and Interface in: Implement the following hierarchy:**

**a. Create a class Student with a variable rollNo and methods getRollNo() and setRollNo().**

**b. Create a class Test that inherits Student and has variables sub1 and sub2 with methods getMarks() and setMarks().**

**c. Create an interface Sports with a variable sMarks and a method set().**

**d. Create a class Result that inherits Test and implements the Sports interface. It should display the marks.**

**e. Demonstrate the functionality of these classes in a test application.**

**Answer:**

import java.util.\*;

class student{

int rollNo;

String name;

void setRollNo(int rollNo){

this.rollNo = rollNo;

}

int getRollNo(){

return rollNo;

}

void setName(String name){

this.name = name;

}

String getName(){

return name;

}

}

class Test extends student{

int sub1;

int sub2;

void setMarks(int sub1, int sub2){

this.sub1 = sub1;

this.sub2 = sub2;

}

int getMarks1(){

return sub1;

}

int getMarks2(){

return sub2;

}

}

interface Sports {

void setSportsMarks(int marks);

}

class Result extends Test implements Sports{

int sportsMarks;

public void setSportsMarks(int marks) {

this.sportsMarks = marks;

}

void dispalay(){

System.out.println("Roll number: "+ getRollNo());

System.out.println("Name:" + getName());

System.out.println("Marks in subject 1:" + getMarks1());

System.out.println("Marks in subject 2:" + getMarks2());

System.out.println("Sports Marks:" + sportsMarks);

System.out.println("Total Marks:" + (getMarks1()+getMarks2() +sportsMarks));

}

}

public class StudentExamResults {

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

Result r = new Result();

System.out.print("Enter the roll number:");

int rollNo = sc.nextInt();

r.setRollNo(rollNo);

sc.nextLine();

System.out.print("Enter the name:");

String name = sc.nextLine();

r.setName(name);

System.out.print("Enter Marks in subject 1:");

int sub1 = sc.nextInt();

System.out.print("Enter marks in subject 2:");

int sub2 = sc.nextInt();

r.setMarks(sub1, sub2);

System.out.print("Enter the sports Marks:");

int sportsMarks = sc.nextInt();

r.setSportsMarks(sportsMarks);

r.dispalay();

sc.close();

}

}

Output:

