

Name- Vartika Singh

Sap id- 500085957

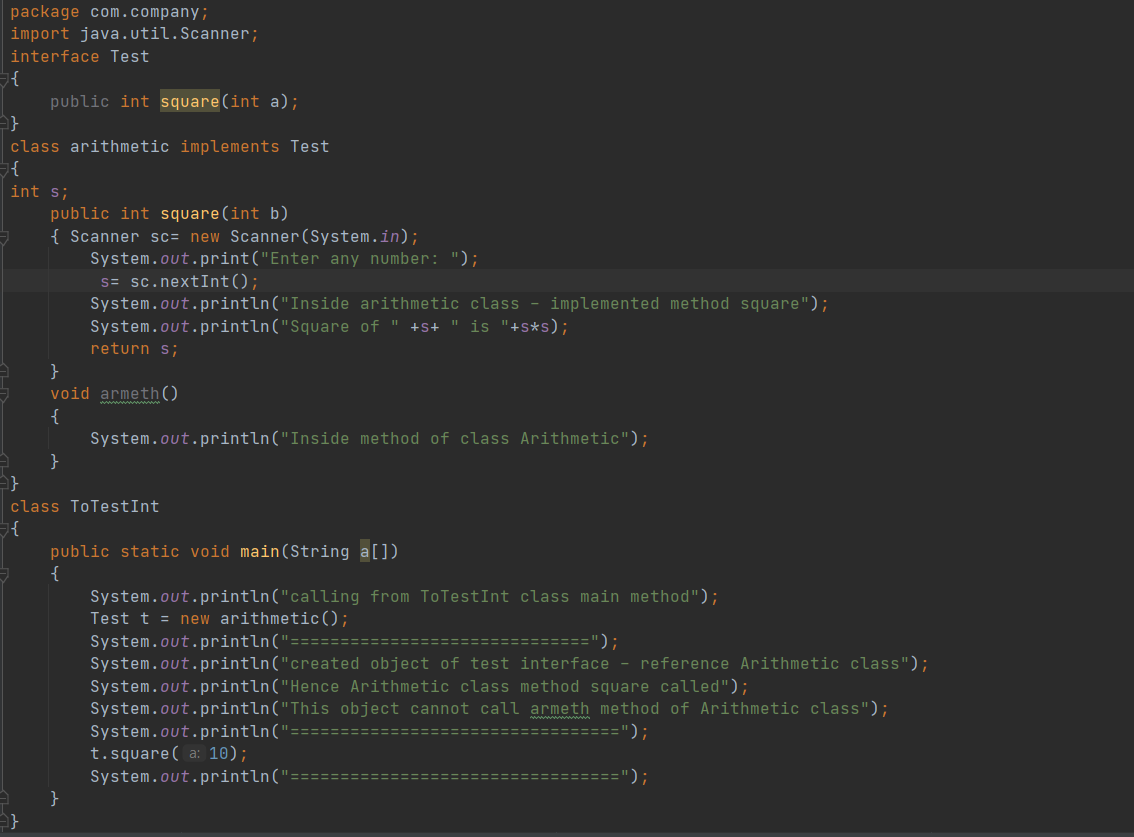
**OBJECT ORIENTED PROGRAMMING**

**LAB EXPERIMENT-4 (INTERFACE)**

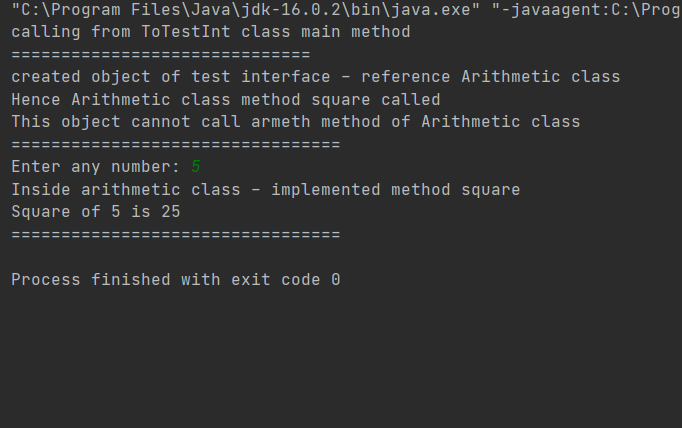
1. **Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt. In this class use the object of arithmetic class.**

**CODE:**

package com.company;  
import java.util.Scanner;  
interface Test  
{  
 public int square(int a);  
}  
class arithmetic implements Test  
{  
int s;  
 public int square(int b)  
 { Scanner sc= new Scanner(System.*in*);  
 System.*out*.print("Enter any number: ");  
 s= sc.nextInt();  
 System.*out*.println("Inside arithmetic class – implemented method square");  
 System.*out*.println("Square of " +s+ " is "+s\*s);  
 return s;  
 }  
 void armeth()  
 {  
 System.*out*.println("Inside method of class Arithmetic");  
 }  
}  
class ToTestInt  
{  
 public static void main(String a[])  
 {  
 System.*out*.println("calling from ToTestInt class main method");  
 Test t = new arithmetic();  
 System.*out*.println("==============================");  
 System.*out*.println("created object of test interface – reference Arithmetic class");  
 System.*out*.println("Hence Arithmetic class method square called");  
 System.*out*.println("This object cannot call armeth method of Arithmetic class");  
 System.*out*.println("=================================");  
 t.square(10);  
 System.*out*.println("=================================");  
 }  
}



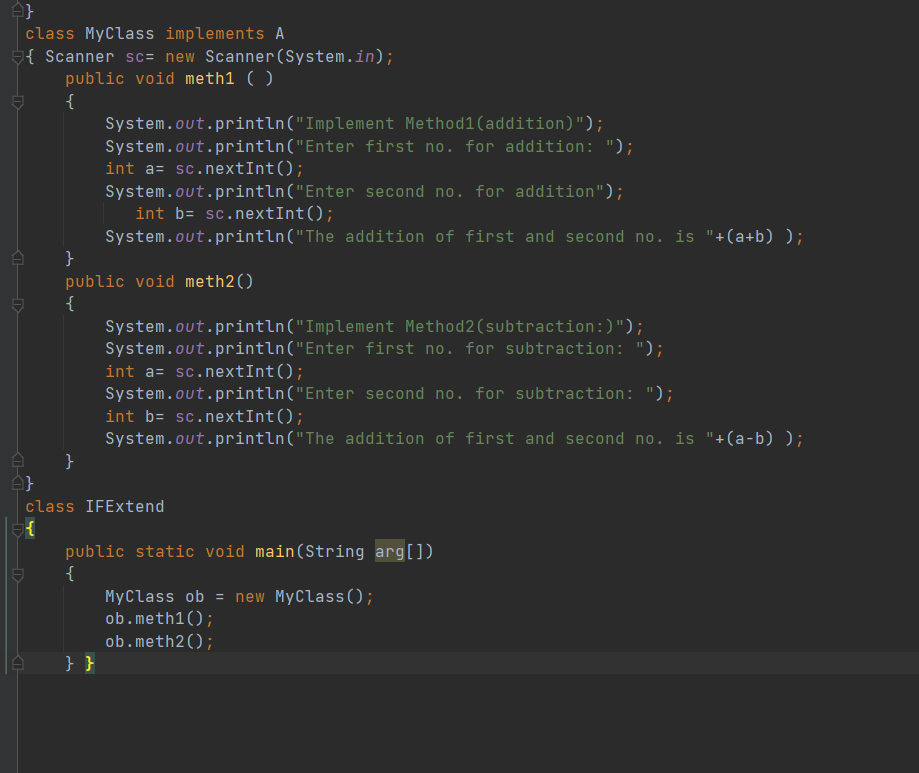
**OUTPUT:**



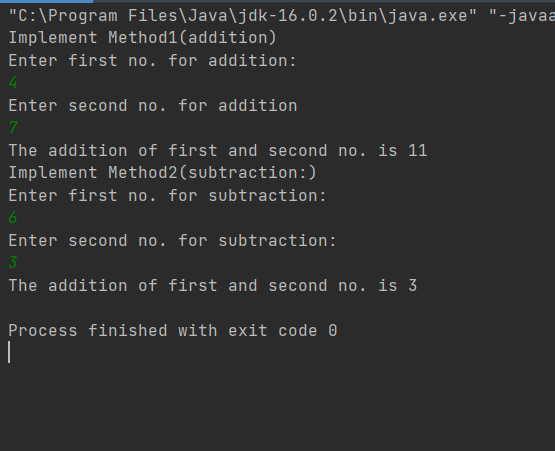
1. **Write a program to create interface A, in this interface we have two method meth1 and meth2. Implements this interface in another class named MyClass.**

**CODE:**

import java.util.Scanner;  
interface A  
{  
 void meth1();  
 void meth2();  
}  
class MyClass implements A  
{ Scanner sc= new Scanner(System.*in*);  
 public void meth1 ( )  
 {  
 System.*out*.println("Implement Method1(addition)");  
 System.*out*.println("Enter first no. for addition: ");  
 int a= sc.nextInt();  
 System.*out*.println("Enter second no. for addition");  
 int b= sc.nextInt();  
 System.*out*.println("The addition of first and second no. is "+(a+b) );  
 }  
 public void meth2()  
 {  
 System.*out*.println("Implement Method2(subtraction:)");  
 System.*out*.println("Enter first no. for subtraction: ");  
 int a= sc.nextInt();  
 System.*out*.println("Enter second no. for subtraction: ");  
 int b= sc.nextInt();  
 System.*out*.println("The addition of first and second no. is "+(a-b) );  
 }  
}  
class IFExtend  
{  
 public static void main(String arg[])  
 {  
 MyClass ob = new MyClass();  
 ob.meth1();  
 ob.meth2();  
 } }



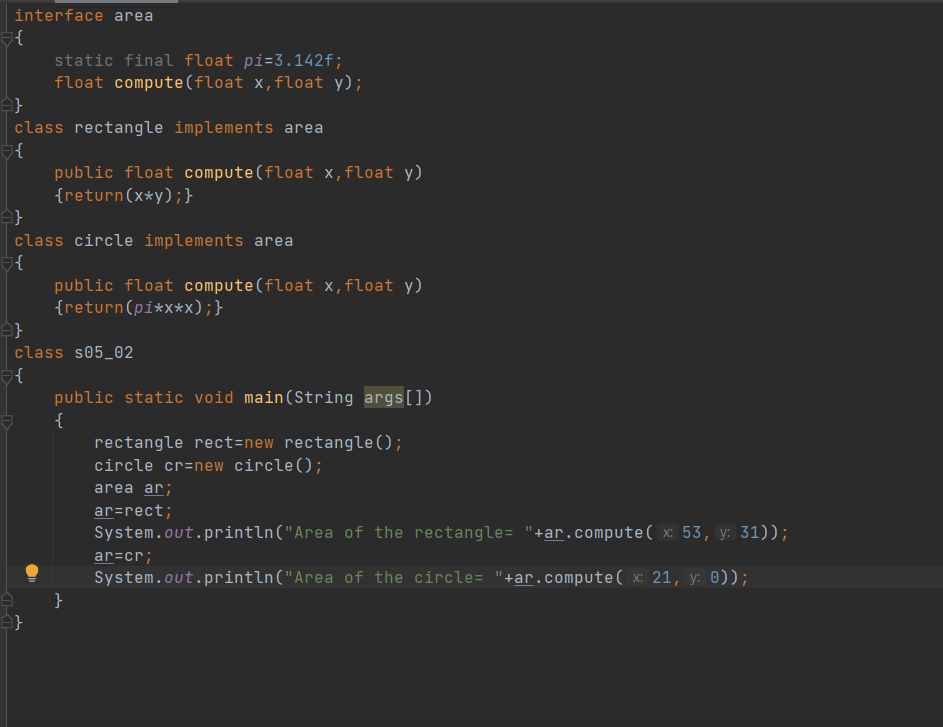
**OUTPUT:**



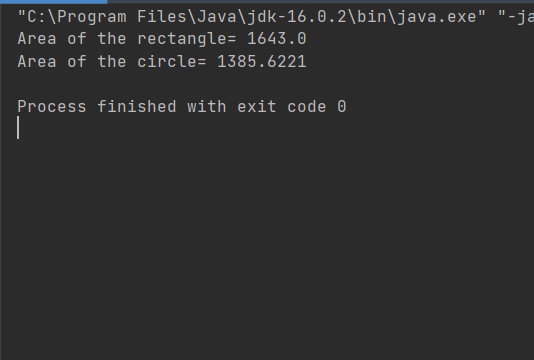
1. **Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.**

**CODE:**

interface area  
{  
 static final float *pi*=3.142f;  
 float compute(float x,float y);  
}  
class rectangle implements area  
{  
 public float compute(float x,float y)  
 {return(x\*y);}  
}  
class circle implements area  
{  
 public float compute(float x,float y)  
 {return(*pi*\*x\*x);}  
}  
class s05\_02  
{  
 public static void main(String args[])  
 {  
 rectangle rect=new rectangle();  
 circle cr=new circle();  
 area ar;  
 ar=rect;  
 System.*out*.println("Area of the rectangle= "+ar.compute(53,31));  
 ar=cr;  
 System.*out*.println("Area of the circle= "+ar.compute(21,0));  
 }  
}



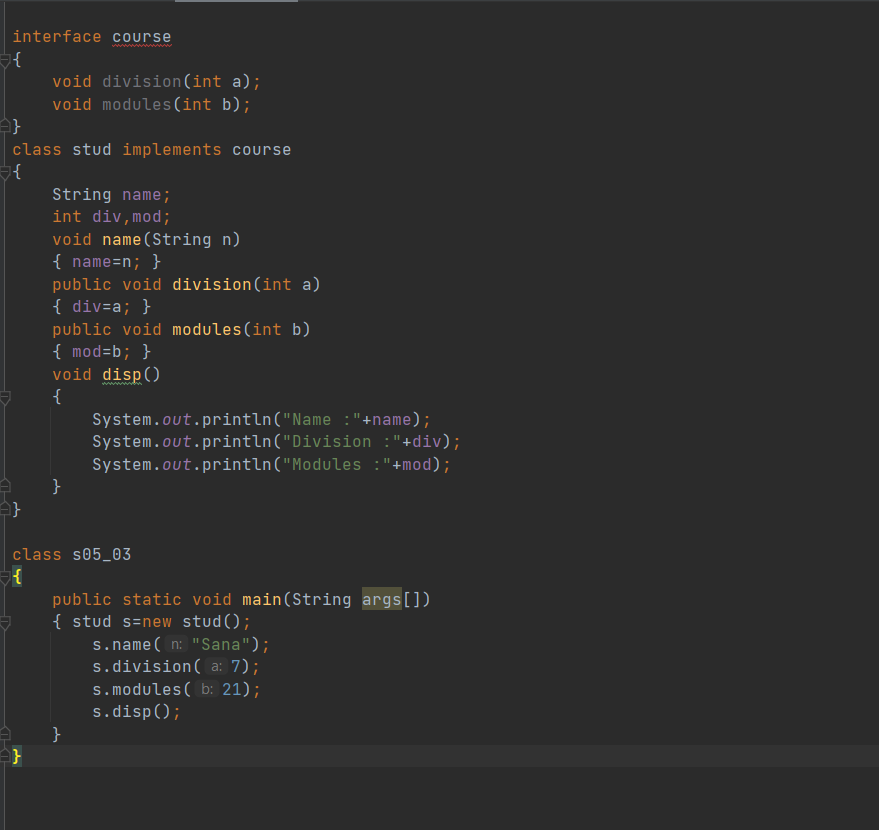
**OUTPUT:**



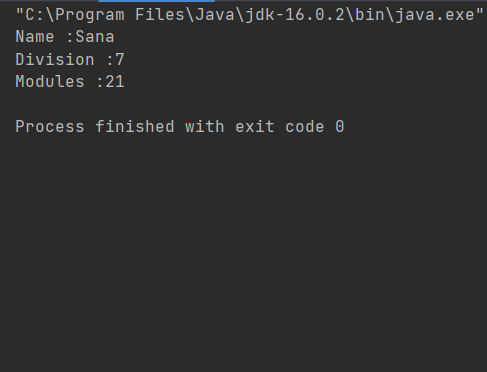
1. **Write a program to create an Interface having two methods division and modules. Create a class, which overrides these methods.**

**CODE:**

interface course  
{  
 void division(int a);  
 void modules(int b);  
}  
class stud implements course  
{  
 String name;  
 int div,mod;  
 void name(String n)  
 { name=n; }  
 public void division(int a)  
 { div=a; }  
 public void modules(int b)  
 { mod=b; }  
 void disp()  
 {  
 System.*out*.println("Name :"+name);  
 System.*out*.println("Division :"+div);  
 System.*out*.println("Modules :"+mod);  
 }  
}  
  
class s05\_03  
{  
 public static void main(String args[])  
 { stud s=new stud();  
 s.name("Sana");  
 s.division(7);  
 s.modules(21);  
 s.disp();  
 }  
}

****

**OUTPUT:**

****

1. **Write program to create an interface StackInterface having methods push (), pop () and display (). StackClass implements StackInterface. Class StackClass contains the main method which is having a switch case for selecting the particular operation of the stack.**

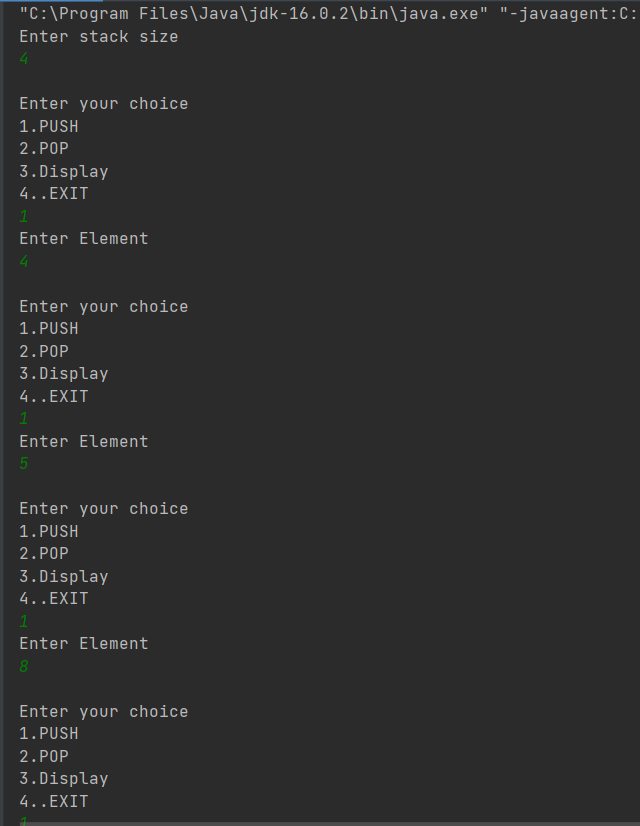
**CODE:**

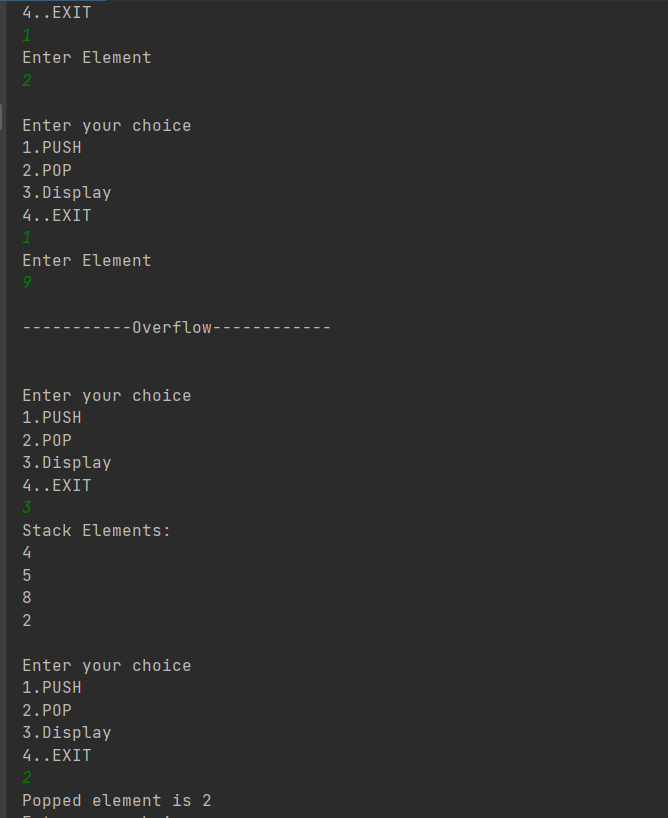
import java.io.\*;  
import java.util.Scanner;  
class stack {  
 static int *ch*;  
 int element, maxsize, top;  
 int[] st;  
 public stack() {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter stack size");  
 maxsize = sc.nextInt();  
 st = new int[maxsize];  
 top = -1;  
 }  
 public void push(int element) {  
 if(top ==maxsize-1) {  
 System.*out*.println("\n-----------Overflow------------\n");  
 } else {  
 try {  
 st[++top] = element;  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println(e);  
 }  
 }  
 }  
 public int pop() {  
 if (top == -1) {  
 System.*out*.println("\n----------UnderFlow-----------\n");  
 return (-1);  
 }  
 else {  
 System.*out*.printf("Popped element is " +(st[top--]));  
 return 0;  
 }  
 }  
  
 public void display(int[] st, int max\_size) {  
 int i;  
 System.*out*.println("Stack Elements:");  
 for (i = 0; i <= max\_size; i++)  
 System.*out*.println(st[i]);  
 new myStack();  
 }  
}  
 class myStack {  
 static int *ch*;  
  
 public static void main(String[] args) {  
 stack obj = new stack();  
 while (true) {  
 System.*out*.println("\nEnter your choice\n1.PUSH\n2.POP\n3.Display\n4..EXIT");  
 Scanner integer = new Scanner(System.*in*);  
 *ch* = integer.nextInt();  
 switch (*ch*) {  
 case 1:  
 System.*out*.println("Enter Element");  
 obj.element = integer.nextInt();  
 obj.push(obj.element);  
 break;  
 case 2:  
 obj.pop();  
 break;  
 case 3:  
 obj.display(obj.st, obj.top);  
 break;  
 case 4:  
 System.*exit*(0);  
 default:  
 System.*out*.println("Wrong option");  
 }  
 }  
 }  
}

****

****

**OUTPUT:**

****

**** 