# ASSIGNMENT -2

ROLLNO -27566

HITESH KUMAR

1. Java program to print the following spiral pattern on the console

**public class** SpiralPatternExample1

{

//defining method to print the spiral pattern or matrix

**static void** printSpiralPattern(**int** size)

{

//create two variables row and col to traverse rows and columns

**int** row = 0, col = 0;

**int** boundary = size - 1; **int** sizeLeft = size - 1; **int** flag = 1;

//variables r, l, u and d are used to determine the movement

// r = right, l = left, d = down, u = upper

**char** move = 'r';

//creating a 2D array for matrix **int**[][] matrix =**new int** [size][size]; **for** (**int** i = 1; i < size \* size + 1; i++)

{

//assigning values matrix[row][col] = i;

//switch-case to determine the next index

**switch** (move)

{

//if right, go right

**case** 'r':

col += 1;

**break**;

//if left, go left

**case** 'l':

col -= 1;

**break**;

//if up, go up

**case** 'u':

row -= 1;

**break**;

//if down, go down

**case** 'd':

row += 1;

**break**;

}

//checks if the matrix has reached the array boundary

**if** (i == boundary)

{

//adds the left size for the next boundary boundary = boundary + sizeLeft;

//decrease the size left by 1, if 2 rotations have been made

**if** (flag != 2)

{

flag = 2;

}

**else**

{

flag = 1;

sizeLeft -= 1;

}

//switch-case to rotate the movement

**switch** (move)

{

//if right, rotate to down

**case** 'r':

move = 'd';

**break**;

// if down, rotate to left

**case** 'd':

move = 'l';

**break**;

// if left, rotate to up

**case** 'l':

move = 'u';

**break**;

// if up, rotate to right

**case** 'u':

move = 'r';

**break**;

}

}

}

//printing the spiral matrix or pattern

//outer for loop for rows

**for** (row = 0; row < size; row++)

{

//inner for loop for columns

**for** (col = 0; col < size; col++)

{

**int** n = matrix[row][col];

**if**(n < 10) System.out.print(n +" ");

**else**

System.out.print(n +" ");

}

System.out.println();

}

}

//driver Code

**public static void** main(String args[])

{

//size of the array?s row and column

**int** size = 5;

System.out.println("Spiral Matrix or Pattern is: \n");

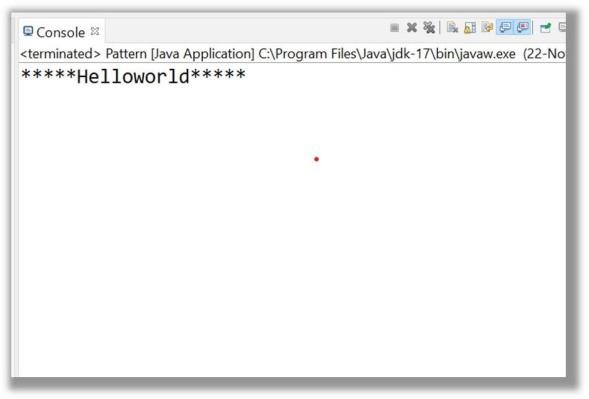
//calling the method that prints the spiral pattern or matrix printSpiralPattern(size);

}

}

Output:





1. Java program to print the following pattern

\*\*\*\*\*\*\*\*\*\*Hello\*\*\*\*\*\*\*\*\*\*

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=10;

**for**(**int** i=0;i<n/2;i++) { System.***out***.print("\*");

}

System.***out***.print("Helloworld");

**for**(**int** i=n/2;i<n;i++) { System.***out***.print("\*");

}

}

}

Output:

1. Java program to print the following pattern 2

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=10;

**for**(**int** i=n;i>6;--i) {

**for**(**int** j=1;j<=i;j++) { System.***out***.print("\*");

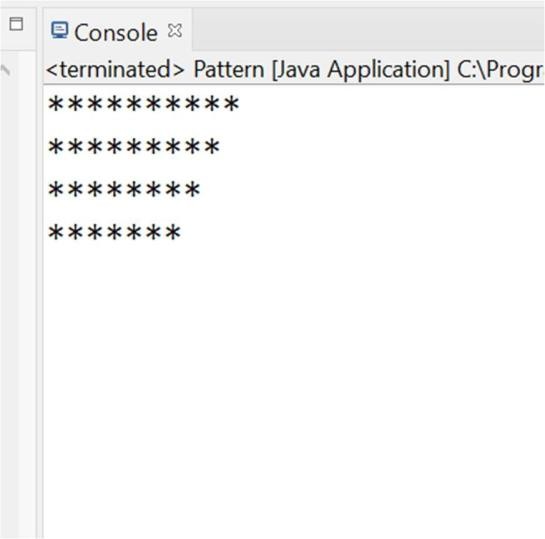
}

System.***out***.println();

}

}

}



1. Java program to print the following pattern 3

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=5;

**for**(**int** i=1;i<=5;i++) {

**for**(**int** j=1;j<=i;j++) { System.***out***.print("\*");

}

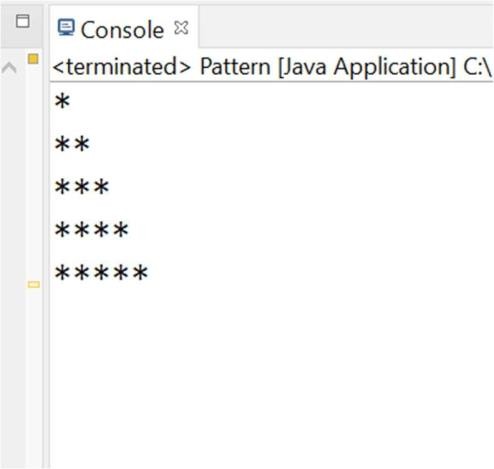
System.***out***.println();

}

}

}

Output:



1. Java program to print the following pattern 4 1

12

123

1234

12345

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=5;

**for**(**int** i=1;i<=5;i++) {

**for**(**int** j=1;j<=i;j++) { System.***out***.print(j);

}

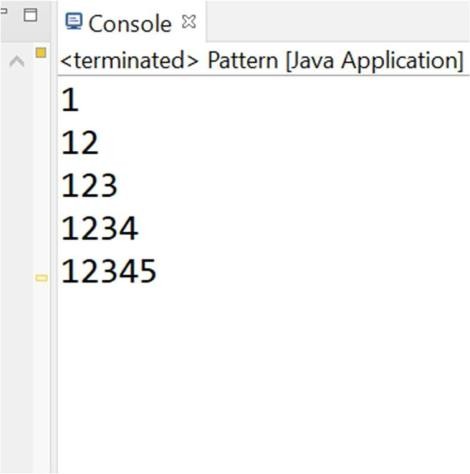
System.***out***.println();

}

}

}

Output:



1. Java program to print the following pattern 5 11111

22222

33333

44444

55555

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=5;

**for**(**int** i=1;i<=5;i++) {

**for**(**int** j=1;j<=5;j++) { System.***out***.print(i);

}

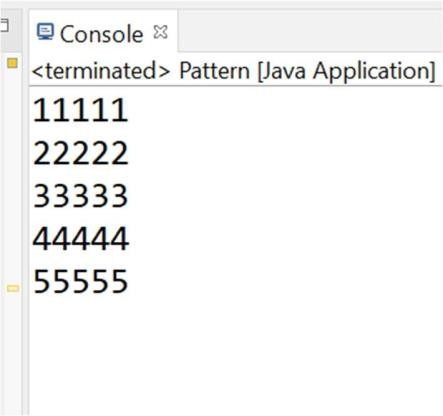
System.***out***.println();

}

}

}

Output:



1. Java program to print the following pattern on the console 2 12345

12345

12345

12345

12345

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=5;

**for**(**int** i=1;i<=5;i++) {

**for**(**int** j=1;j<=5;j++) { System.***out***.print(j);

}

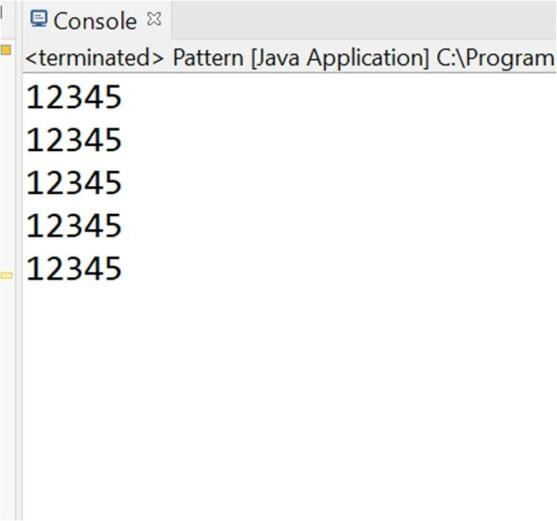
System.***out***.println();

}

}

}

Output:



1. Java program to print the following pattern on the console 3 11111

22111

### 33311

44441

### 55555

**public class** Pattern {

**public static void** main(String[] args) {

**int** n=5;

**for**(**int** i=1;i<=5;i++) {

**for**(**int** j=1;j<=i;j++) { System.***out***.print(i);

}

**for**(**int** k=n-i;k>=1;k--) {

System.***out***.print("1");

}

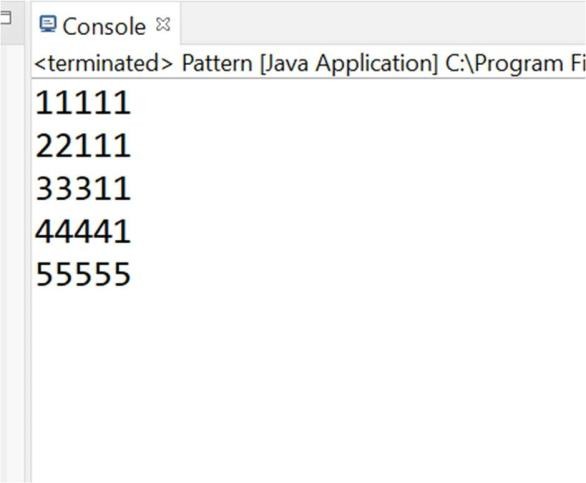
System.***out***.println();

}

}

}

Output:



Java Programs

## Write a java program to create objects using different methods of creation of the object.

**public class** Display {

**public void** dis(**int** x) { System.***out***.println("====dis(int x)======"); System.***out***.println("the value of x is"+x);

}

}

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

**public class** Question1 {

String name = "greatforgreat";

**public static void** main(String[] args) **throws** ClassNotFoundException,InstantiationException,IllegalAcces sException {

// method using new operator Scanner s = **new** Scanner(System.***in***);

//String class object

String str = "HElloworld"; System.***out***.println(str.toString());

//new instance method

Class c = Class.*forName*("javabasicprogram.Display");

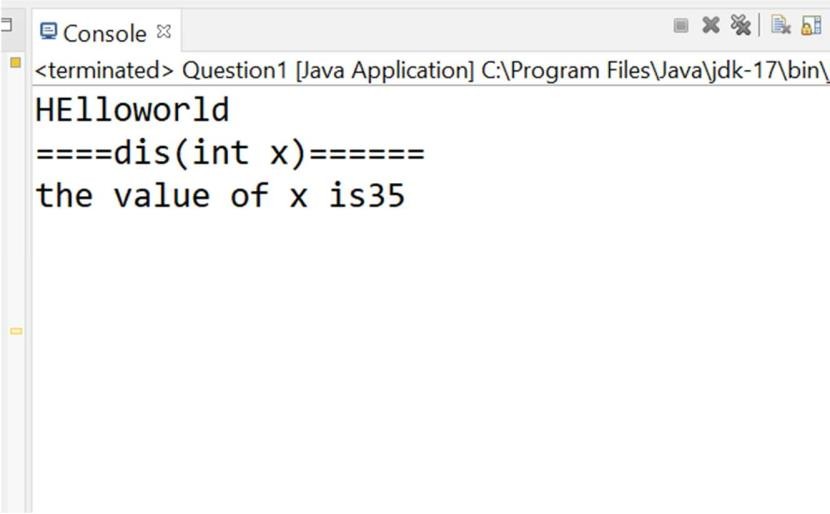
javabasicprogram.Display ob = (javabasicprogram.Display)c.~~newInstance~~();

ob.dis(35);

}

}

Output:



### Write a program to show the nesting of classes in java using five classes.

**public class** Nesting {

#### public class a{

**public void** m1() { System.***out***.println("class 1");

}

#### public class b{

**public void** m2() { System.***out***.println("class 2");

}

#### public class c{

**public void** m3() { System.***out***.println("class 3");

}

#### public class d{

**public void** m4() { System.***out***.println("class 4");

5");

}

#### public class e{

**public void** m5() { System.***out***.println("class

}

}

}

}

}

}

}

**import** javabasicprogram.Nesting;

**public class** Nesting2 {

**public static void** main(String[] args) { Nesting n = **new** Nesting();

Nesting.a obj1 = n.**new** a(); obj1.m1();

Nesting.a.b obj2 = obj1.**new** b(); obj2.m2();

Nesting.a.b.c obj3 = obj2.**new** c(); obj3.m3();

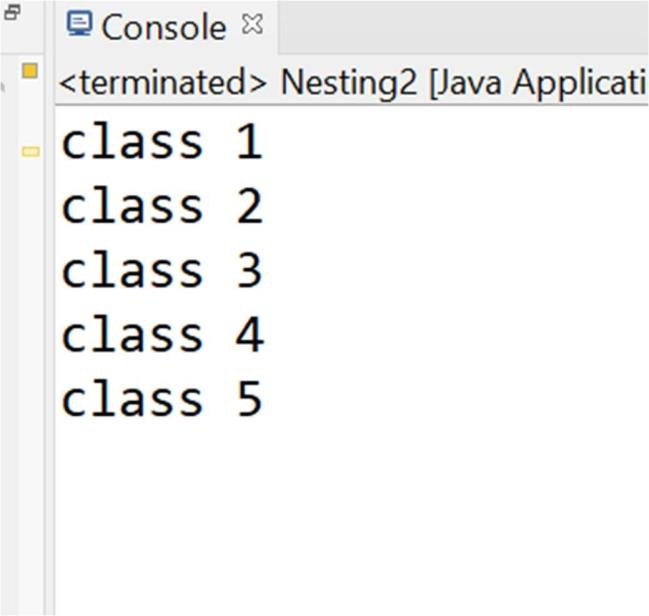
Nesting.a.b.c.d obj4 = obj3.**new** d(); obj4.m4();

Nesting.a.b.c.d.e obj5 = obj4.**new** e(); obj5.m5();

}

}

Output:



### Write a program in java to show the different prototypes of the main method used in the classes.

**package** p2;

**class** DemoStaticBlock

{

**static** //static block

{

System.***out***.println("Static block");

}

}

**class** Q3

{

**public static void** main(**int** a) //overloaded main method

{

System.***out***.println(a);

}

**public static void** main(String args[])

{

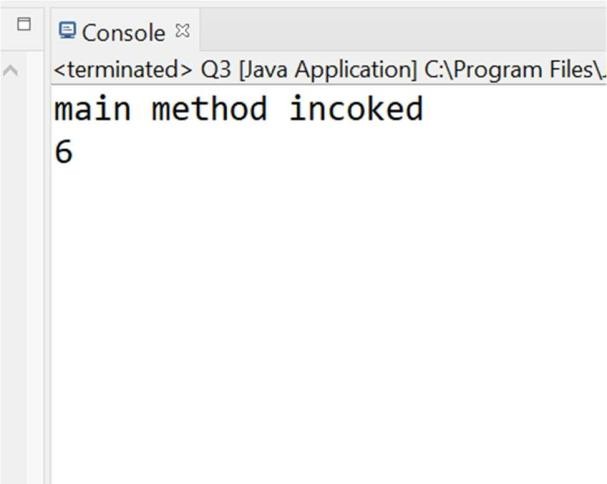
System.***out***.println("main method incoked");

*main*(6);

}

}

Output:



# Write a program in java to show abstraction.

**abstract class** Bike{

**abstract void** run();

}

**class** Honda4 **extends** Bike{

**void** run(){System.***out***.println("running safely");}

}

#### public class Q4

{

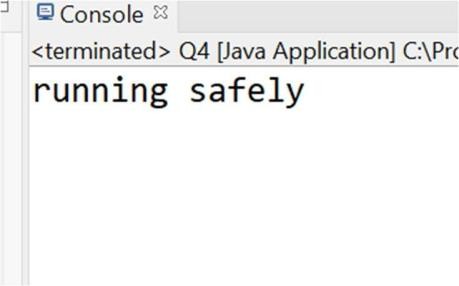
**public static void** main(String[] args) { Bike obj = **new** Honda4();

obj.run();

}

}

Output:



1. Write a program to show the different types of constructors available in java and their scope.

**class** Bike1{

//creating a default constructor Bike1(){System.***out***.println("Bike is created");} }

**class** Student3{

**int** id; String name;

//method to display the value of id and name

**void** display(){System.***out***.println(id+" "+name);} }

**class** Student4{ **int** id; String name;

//creating a parameterized constructor Student4(**int** i,String n){

id = i; name = n;

}

//method to display the values

**void** display(){System.***out***.println(id+" "+name);}}

#### public class Q5

{

**public static void** main(String[] args) { Bike1 b=**new** Bike1();

Student3 s1=**new** Student3(); Student3 s2=**new** Student3();

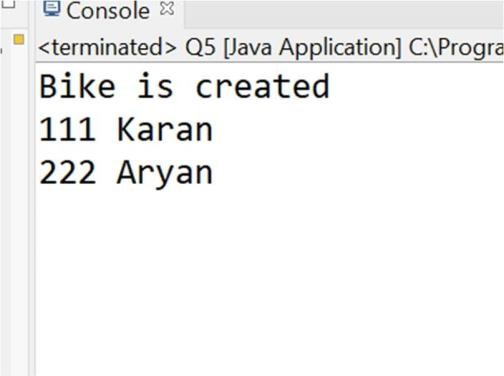
Student4 s3 = **new** Student4(111,"Karan"); s3.display();

Student4 s4 = **new** Student4(222,"Aryan"); s4.display();

}

}

Output:



1. Write a program in java to show the different access specifiers available in java.

**public class** Test { **private int** k = 600; **private class** Sub{

**public void** m2(**int** x) { System.***out***.println("=======innerclass =========="); System.***out***.println("the value of x :"+x);

}

}

**public void** m1() { System.***out***.println("private Variable"); System.***out***.println("the value of k :"+k);

}

**private** Test() {}

**private static** Test *ob* = **null**; **public void** acess() {

Sub sb = **new** Sub(); sb.m2(67);

}

**public static** Test getReference() {

**if**(*ob* == **null**) {

*ob* = **new** Test();

}

**return** *ob*;

}

}

**import** p1.Test;

**class** Demoprivate1 {

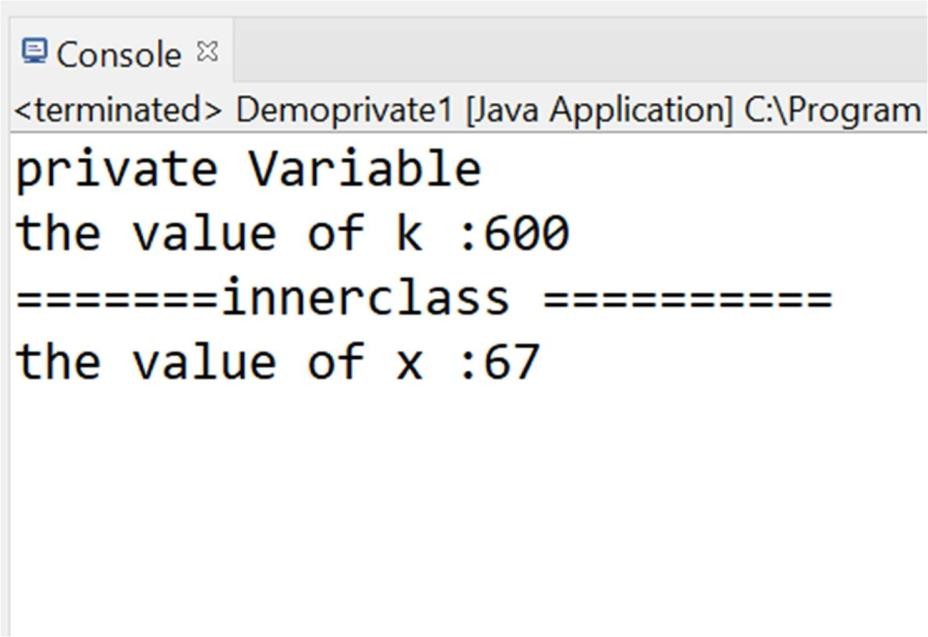
**public static void** main(String[] args) { Test ob = Test.*getReference*(); ob.m1();

ob.acess();

}

}

Output:



### Write a program in java to show the access specifiers with instance variables, local

variables, anonymous blocks, and static blocks.

**class** CTest2

{

**int** a = 10;

**static int** *b* = 20;

#### static{

System.***out***.println("------static block"); System.***out***.println("the value of b is"+*b*);

}

{

--");

}

System.***out***.println("------instance block---------

System.***out***.println("the value of a is"+a); System.***out***.println("the value of b is"+*b*);

");

CTest2(){

System.***out***.println("------Constructor-----------

System.***out***.println("the value of a is"+a); System.***out***.println("the value of b is"+*b*);

}

**void** dis(){

System.***out***.println("------instance method--------

---");

}

System.***out***.println("the value of a is"+a); System.***out***.println("the value of b is"+*b*);

}

**class** Con2

{

**public static void** main(String[] args)

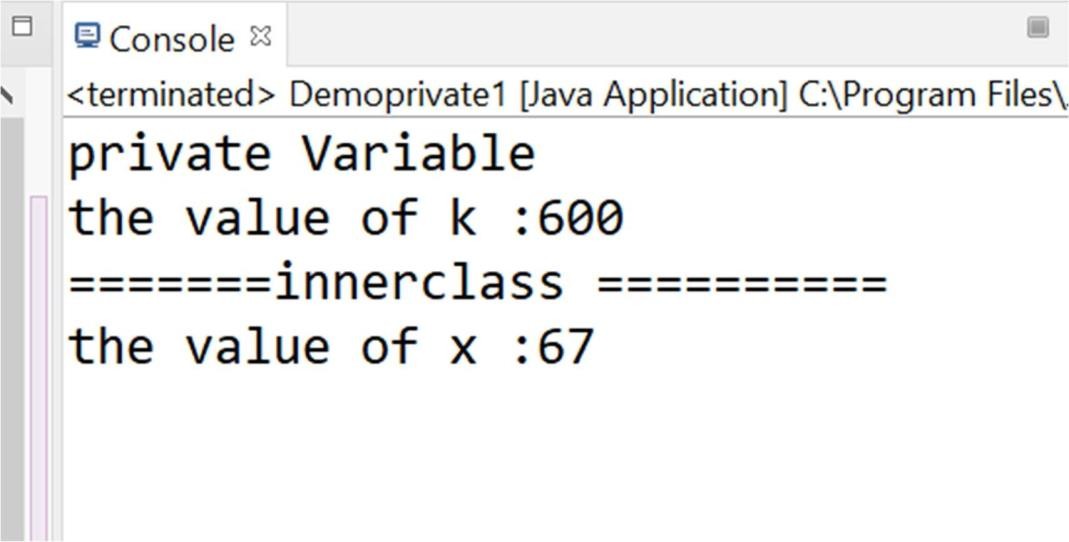
{

CTest2 ob = **new** CTest2(); ob.dis();

}

}

Output:



1. Write a program in java to show the static import of packages, System classes, Print

stream classes, Object classes, etc.

**package** p1;

**import static** java.lang.Math.\*;

**public class** Calculate {

**public static double** cal(**double** x) {

**return** *sqrt*(x);

}

}

**package** p2;

**import** java.util.Scanner;

**import** java.io.\*;

**import static** p1.Calculate.\*;

**public class** Smainclass {

**public static void** main(String[] args) { PrintStream stream = **new**

PrintStream(System.***out***);

String str = **new** String("welcome to java"); stream.print(str);

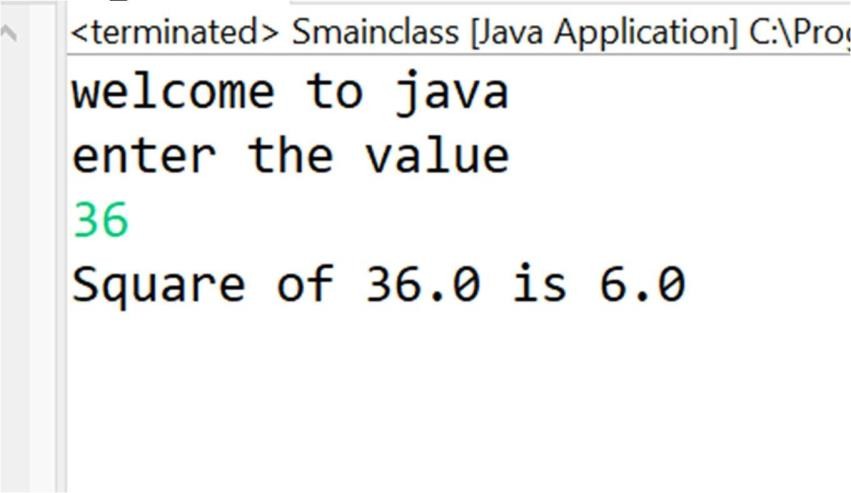
Scanner s = **new** Scanner(System.***in***); System.***out***.println("\nenter the value"); **double** x = s.nextDouble();

**double** r = *cal*(x); System.***out***.println("Square of "+x+" is "+r); s.close();

}

}

Output:



### Write a program in java to show the compile time and run time polymorphism.

**class** Shapes {

**public void** area() {

System.***out***.println("The formula for area of ");

}

}

");

**class** Triangle **extends** Shapes {

**public void** area() {

System.***out***.println("Triangle is ½ \* base \* height

}

}

**class** Circle **extends** Shapes {

**public void** area() {

System.***out***.println("Circle is 3.14 \* radius \* radius ");

}

}

**class** Main {

**public static void** main(String[] args) { Shapes myShape = **new** Shapes(); // Create a

Shapes object

Shapes myTriangle = **new** Triangle(); // Create a Triangle object

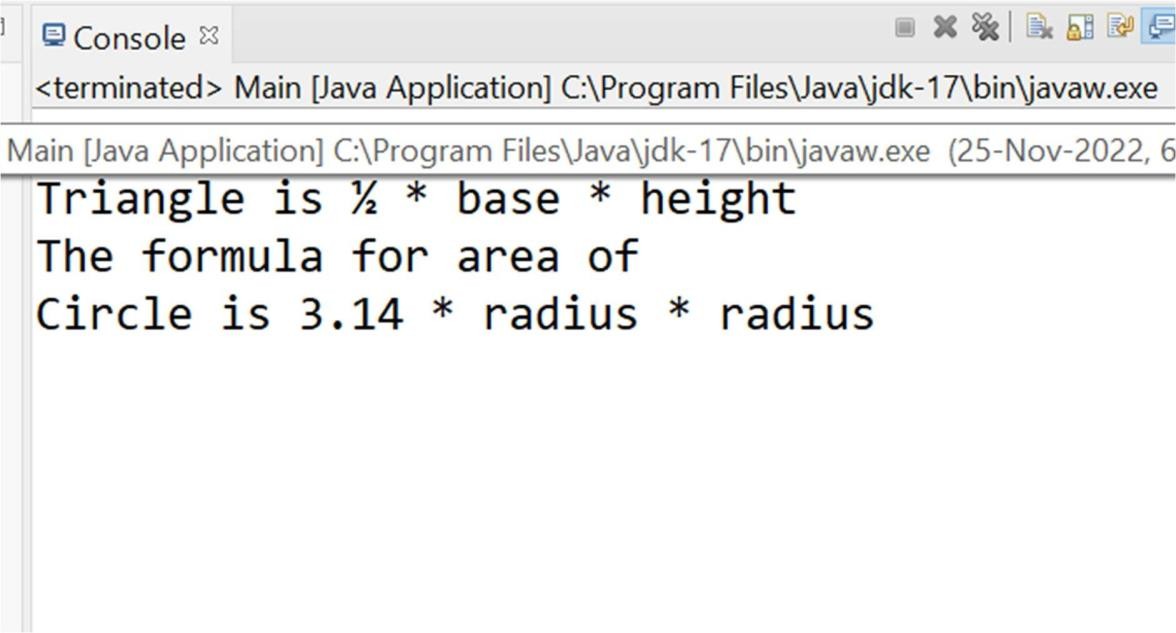
Shapes myCircle = **new** Circle(); // Create a Circle object

myShape.area(); myTriangle.area(); myShape.area(); myCircle.area();

}

}

Output:



### Write a program in java to show Inheritance and its types.

Package p1;

**public interface** Itest {

**public abstract void** Hello(**int** x);

}

Package p1;

**public class** IClass **implements** Itest{

**public static void** main(String[] args) { Itest ob = (**int** x)->{

System.***out***.println("lamdab Expression(int x)"); System.***out***.println("the value of x is "+x);

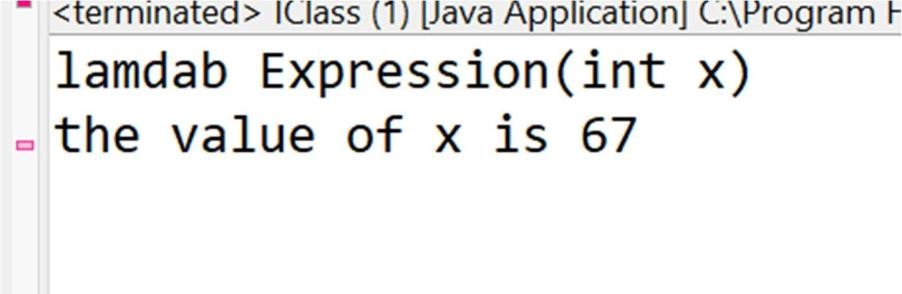
};

ob.Hello(67);

}

}

Output:



Multiple inheritance

**package** src.p1;

**public interface** Interface1 {

**public abstract int** compare(**int** x,**int** y);

}

package src.p2; import src.p1.\*; import java.util.\*;

public class DemoInterface3 {

public static void main(String[] args) { Scanner s = new Scanner(System.in); System.out.println("enter the value 1:"); int v1 = s.nextInt(); System.out.println("enter the value 2 :"); int v2 = s.nextInt(); System.out.println("choice"); switch(s.nextInt()) {

case 1:

Interface1 g = (int x, int y)->{

if(x>y) return x; else return y;

};

int r1 = g.compare(v1,v2); System.out.println("Greater value is"+r1); break;

case 2:

Interface1 small = (int x,int y)->{ if(x<y) return y;

else return x;

};

int r2 = small.compare(v1,v2); System.out.println("Greater value is"+r2); break;

default:

System.out.println("Invalid choice");

}

s.close();

}

}

Output:

