import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("size n m");

int n=sc.nextInt(),m=sc.nextInt();

System.out.println("1st matrix");

int[][] a=new int[n][m],b=new int[n][m],c=new int[n][m];

for(int i=0;i<n;i++)

for(int j=0;j<m;j++)

a[i][j]=sc.nextInt();

System.out.println("2st matrix");

for(int i=0;i<n;i++)

for(int j=0;j<m;j++)

b[i][j]=sc.nextInt();

for(int i=0;i<n;i++)

for(int j=0;j<m;j++)

c[i][j]=a[i][j]+b[i][j];

System.out.println("result matrix");

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

for(int j=0;j<m;j++)

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

System.out.println();}

}

}

Output:

size n m

2 2

1st matrix

1 1

1 1

2st matrix

1 1

1 1

result matrix

2 2

2 2

2

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("size n m");

int n=sc.nextInt(),m=sc.nextInt();

System.out.println("1st matrix");

int[][] a=new int[n][m];

for(int i=0;i<n;i++)

for(int j=0;j<m;j++)

a[i][j]=sc.nextInt();

System.out.println("n1 m1");

int n1=sc.nextInt(),m1=sc.nextInt();

if(m!=n1){

System.out.println("multiplication not possible");

}else{

System.out.println("2st matrix");

int[][] b=new int[n1][m1],c=new int[n][m1];

for(int i=0;i<n1;i++)

for(int j=0;j<m1;j++)

b[i][j]=sc.nextInt();

for(int k=0;k<n;k++)

for(int i=0;i<m1;i++)

for(int j=0;j<n1;j++)

c[k][i]+=a[k][j]\*b[j][i];

System.out.println("result matrix");

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

for(int j=0;j<m1;j++)

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

System.out.println();}}

}

}

Output:

size n m

3 1

1st matrix

4

5

6

n1 m1

1 3

2st matrix

1 2 3

result matrix

4 8 12

5 10 15

6 12 18

3

class Main

{

void sum(int l,long b)

{

System.out.println("Sum is"+(l+b)) ;

}

void sum(int l, int b, int h)

{

System.out.println("Sum is"+(l+b+h));

}

public static void main (String[] args)

{

Main ar = new Main();

ar.sum(8,5);

ar.sum(8,5,1);

}

}

Output:

13

14

4.

import java.util.\*;

public class Main{

int x,y;

Main(){

x=6;

y=1;

System.out.println(x+" "+y);

}

Main(int x,int y){

this.x=x;

this.y=y;

System.out.println(x+" "+y);

}

public static void main(String[] args) {

Main m=new Main();

Main m1=new Main(1,2);

}

}

Output:

6 1

1 2

5.

import java.util.\*;

public class Main{

static int show(int x,int y){

return x+y;

}

public static void main(String[] args) {

System.out.println(show(2,4));

}

}

Output:

1.

Conditional statements help you to make a decision based on certain conditions. These conditions are specified by a set of conditional statements having boolean expressions which are evaluated to a boolean value true or false.

2.

The syntax of the switch statement is:

switch (variable/expression) {

case value1:

// statements of case1

break;

case value2:

// statements of case2

break;

.. .. ...

.. .. ...

default:

// default statements

}

3.

The break statement is primarily used as the exit statement, which helps in escaping from the current block or loop. Conversely, the continue statement helps in jumping from the current loop iteration to the next loop.

4.

A loop statement is a series of steps or sequence of statements executed repeatedly zero or more times satisfying the given condition is satisfied. In programming languages, such as C, C++, Java, and Python, come with “For, While and Do loop” statements.

5.

In 'while' loop the controlling condition appears at the start of the loop.

In 'do-while' loop the controlling condition appears at the end of the loop.

6.

Arrays in Java, as in other languages, are a way to store collections of items into a single unit. The array has some number of slots, each of which holds an individual item. You can add and delete items to those slots as needed. Unlike in other languages, however, arrays in Java are actual objects that can be passed around and treated just like other objects.

The first step in creating an array is creating a variable that will hold the array, just as you would any other variable. Array variables indicate the type of object the array will hold (just as they do for any variable) and the name of the array, followed by empty brackets ([]). The following are all typical array variable declarations:

String difficultWords[];

Point hits[];

int temps[];

7.

A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.

8.

Constructor is a block of code that initializes the newly created object. A constructor resembles an instance method in java but it’s not a method as it doesn’t have a return type

9.

Copy Constructor is a type of constructor which is used to create a copy of an already existing object of a class type. It is usually of the form X (X&), where X is the class name. The compiler provides a default Copy Constructor to all the classes.

10.

Keyword THIS is a reference variable in Java that refers to the current object.

It can be used to refer instance variable of current class

It can be used to invoke or initiate current class constructor

It can be passed as an argument in the method call

It can be passed as argument in the constructor call

It can be used to return the current class instance

11.

Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different

In order to overload a method, the argument lists of the methods must differ in either of these:

1. Number of parameters.
2. Data type of parameters.
3. Sequence of Data type of parameters.

12.

If you declare any variable as static, it is known as a static variable.

The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.

The static variable gets memory only once in the class area at the time of class loading.

13.

Access modifiers are keywords in object-oriented languages that set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the encapsulation of components.

14.

Instance method are methods which require an object of its class to be created before it can be called. To invoke a instance method, we have to create an Object of the class in within which it defined.

Static methods are the methods in Java that can be called without creating an object of class. They are referenced by the class name itself or reference to the Object of that class.

15.

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible)

As you know, a class provides the blueprint for objects; you create an object from a class. Each of the following statements taken from the CreateObjectDemo program creates an object and assigns it to a variable:

Point originOne = new Point(23, 94);

Rectangle rectOne = new Rectangle(originOne, 100, 200);

Rectangle rectTwo = new Rectangle(50, 100);

The first line creates an object of the Point class, and the second and third lines each create an object of the Rectangle class.