**1.Write a java program for Matrix Addition**.

**import** java.util.Scanner

**public** **class** Addmatrix{

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

**int** rowM, colM;

Scanner in = **new** Scanner(System.***in***);

System.***out***.print("Enter Number of Rows and Columns of Matrix : ");

rowM = in.nextInt();

colM = in.nextInt();

**int** M1[][] = **new** **int**[rowM][colM];

**int** M2[][] = **new** **int**[rowM][colM];

**int** resMatrix[][] = **new** **int**[rowM][colM];

System.***out***.print("Enter elements of First Matrix : ");

**for**(**int** i = 0; i < rowM; i++){

**for**(**int** j = 0; j < colM; j++){

M1[i][j] = in.nextInt();

}

}

System.***out***.println("First Matrix : " );

**for**(**int** i = 0; i < rowM; i++){

**for**(**int** j = 0; j < colM; j++){

System.***out***.print(" " +M1[i][j]+"\t");

}

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

}

System.***out***.print("Enter elements of Second Matrix : ");

**for**(**int** i = 0; i < rowM; i++){

**for**(**int** j = 0; j < colM; j++){

M2[i][j] = in.nextInt();

}

}

System.***out***.println("Second Matrix : " );

**for**(**int** i = 0; i < rowM; i++){

**for**(**int** j = 0; j < colM; j++){

System.***out***.print(" " +M2[i][j] + "\t");

}

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

}

// Addition logic

**for**(**int** i = 0; i < rowM; i++){

**for**(**int** j = 0; j < colM; j++){

resMatrix[i][j] = M1[i][j] + M2[i][j];

}

}

// Printing the result matrix

System.***out***.println("Result Matrix : " );

**for**(**int** i = 0; i < resMatrix.length; i++){

**for**(**int** j = 0; j < colM; j++){

System.***out***.print(" " +resMatrix[i][j]+"\t");

}

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

}

}

}

**OUTPUT:**

Enter Number of Rows and Columns of Matrix : 2

3

Enter elements of First Matrix : 2 3 4

2 5 6

First Matrix :

2 3 4

2 5 6

Enter elements of Second Matrix :

4 5 7

1 2 4

Second Matrix :

4 5 7

1 2 4

Result Matrix :

6 8 11

3 7 10

**2.Write a java program for Matrix multiplication.**

**import** java.util.Scanner;

**public** **class** Multi

{

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

{

**int** n;

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Enter the base of squared matrices");

n = input.nextInt();

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

**int**[][] b = **new** **int**[n][n];

**int**[][] c = **new** **int**[n][n];

System.***out***.println("Enter the elements of 1st martix row wise \n");

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

{

**for** (**int** j = 0; j < n; j++)

{

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

}

}

System.***out***.println("Enter the elements of 2nd martix row wise \n");

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

{

**for** (**int** j = 0; j < n; j++)

{

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

}

}

System.***out***.println("Multiplying the matrices...");

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

{

**for** (**int** j = 0; j < n; j++)

{

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

{

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

}

}

}

System.***out***.println("The product is:");

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

{

**for** (**int** j = 0; j < n; j++)

{

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

}

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

}

input.close();

}

}

**OUTPUT:**

Enter the base of squared matrices

2

Enter the elements of 1st martix row wise

2 2

3 3

Enter the elements of 2nd martix row wise

4 4

5 5

Multiplying the matrices...

The product is:

18 18

27 27

**3.Write a java program to demonstrate method overloading?**

Class Main{

void method(int a, int b) {

System.out.println(&quot;Multiplication of&quot; +a+&quot;and&quot;+b+&quot; is=&quot; + (a \*

b));

}

void method(int a, int b, int c) {

System.out.println(&quot;Multiplication of&quot; +a+&quot;,&quot;+b+&quot;and&quot;+c+&quot; is=&quot;

+ (a \* b \* c));

}

}

public class Multiply {

public static void main(String args[]) {

Main m = new Main();

m.mul(2, 4);

m.mul(5, 3, 2);

}

}

**Output:**

Multiplication of2and4 is=8

Multiplication of5,3and2 is=30

**4. Write a java program to create a class Point with two data members**

x&amp;y.Include all constructors and display().?

class Demo{

int x;

int y;

Demo(int x,int y){

x = 10;

y = 20;

}

public void display(){

System.out.println(&quot;sum of&quot;+x+&quot; and &quot;+y+&quot; is: &quot;+(x+y));

}

public static void main(String args[]){

Demo d1 = new Demo();

d1.display();

}

}

**Output:**

sum of10 and 20 is:30

**5.Write a java program to demostrate method overloading.**

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

**class** Methodoverloading {

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

*display*();

}

**static** **void** display() {

System.***out***.println("Java is programming language.");

}

}

**OUTPUT:**

Java is programming language.

**1.What is conditional statement?**

A **conditional statement** is a **statement** that computer programming language used to decide which code has to be run when the true condition is met or which code has not to be run when the true condition is not met.

**2.Write a syntax of switch case statement.**

switch(variable or an integer expression)

{

Case constant:

;

//java code

Case constant:

;

default:

;

}

**3.write the difference between break and continue statement.**

Break:

The break statement has two forms: labeled and unlabeled. You saw the unlabeled form in the previous discussion of the switch statement. You can also use an unlabeled break to terminate a for, while, or do-while loop [...]

An unlabeled break statement terminates the innermost switch, for, while, or do-while statement, but a labeled break terminates an outer statement.

**Continue:**

The continue statement skips the current iteration of a for, while , or do-while loop. The unlabeled form skips to the end of the innermost loop's body and evaluates the boolean expression that controls the loop. [...]

A labeled continue statement skips the current iteration of an outer loop marked with the given label.

**4.** **what is looping statement in java?**

A **loop statement** is a series of steps or sequence of **statements** executed repeatedly zero or more times satisfying the given condition is satisfied. **Loop statements** in programming languages, such as assembly languages or PERL make use of LABEL's to execute the **statement** repeatedly.

**5.Write a difference between while and do while statement.**

If the given condition is false then in **while** loop it is terminated immediately, but in **do while** the statement is executed then the loop is terminated. The minimum number of times a **while** loop can run is ZERO times where as a **do while** loop runs for a minimum of ONE time even if the condition is false.

**6.what is array?How it is created?**

Arrays are objects which store multiple variables of the same type. It can

hold primitive types as well as object references. In fact most of the collection

types in Java which are the part of java.util package use arrays internally in their

functioning. Since Arrays are objects, they are created during runtime .The array

length is fixed.

Features of Array

* Arrays are objects
* They can even hold the reference variables of other objects
* They are created during runtime
* They are dynamic, created on the heap
* The Array length is fixed

In Java, here is how we can declare an array.

dataType[] arrayName

* dataType - it can be primitive data types like int,char,double,byte, etc.

or Java objects

* arrayName - it is an identifier.

**7. What is Class?**

A Class is a group of objects which have common properties. It is a

template or blueprint from which objects are created. It is a logical entity. It can&#39;t

be physical.

A class in Java can contain:

* Fields
* Methods
* Constructors
* Blocks
* Nested class and interface

**8. What is Constructor?**

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. In short constructor and method are different(More

on this at the end of this guide). People often refer constructor as special type of

method in Java.

**9. What is the use of copy constructor?**

A copy constructor in a Java class is a constructor that creates an object using

another object of the same Java class.

That&#39;s helpful when we want to copy a complex object that has several fields, or

when we want to make a deep copy of an existing object.

**10. What is the use of this keyword?**

The this keyword refers to the current object in a method or constructor.

The most common use of the this keyword is to eliminate the confusion

between class attributes and parameters with the same name (because a class

attribute is shadowed by a method or constructor parameter).

this can also be used to:

* Invoke current class constructor
* Invoke current class method
* Return the current class object
* Pass an argument in the method call
* Pass an argument in the constructor call

**11. What is method overloading?**

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. It is

similar to constructor overloading in Java, that allows a class to have more than

one constructor having different argument lists.

When I say argument list it means the parameters that a method has: For

example the argument list of a method add(int a, int b) having two parameters is

different from the argument list of the method add(int a, int b, int c) having three

parameters.

**12.What is Static variable?**

Static variable in Java is variable which belongs to the class and

initialized only once at the start of the execution. It is a variable which belongs to

the class and not to object(instance ). Static variables are initialized only once, at

the start of the execution. These variables will be initialized first, before the

initialization of any instance variables.

* A single copy to be shared by all instances of the class
* A static variable can be accessed directly by the class name and doesn’t

need any object

**13.What is access modifier?**

A Java access modifier specifies which classes can access a given class

and its fields, constructors and methods. Access modifiers can be specified

separately for a class, its constructors, fields and methods. Java access modifiers

are also sometimes referred to in daily speech as Java access specifiers, but the

correct name is Java access modifiers. Classes, fields, constructors and methods

can have one of four different Java access modifiers:

* private
* default (package)
* protected
* public

**14. Write the difference between instance and static methods**?

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. What is object?How it is created?**

It is the basic unit of Object Oriented Programming and it represents the

real life entities.

Real-life entities share two characteristics : they all have attributes and behavior.

An object consists of:

State: It is represented by attributes of an object. It also shows properties of an

object.

Behavior: It is represented by methods of an object. It shows response of an

object with other objects.

Identity: It gives a unique name to an object. It also grants permission to one

object to interact with other objects

Objects are created in 3 ways:

 using new keyword

 using new Instance

 using clone method