Assignment-2

**1.Write a java program for matrix Addition?**

**import** java.util.Scanner;

**public class** AddMatrices

{

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

**int** m, n, p, q;

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

System.***out***.println("Enter no.of rows and col");

m = sc.nextInt();

n = sc.nextInt();

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

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

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

System.***out***.println("Enter the first matrix");

**for** (p = 0; p < m; p++)

**for** (q = 0; q< n; q++)

a[p][q] = sc.nextInt();

System.***out***.println("Enter the second matrix");

**for** (p = 0 ; p< m; p++)

**for** (q = 0 ; q < n; q++)

b[p][q] = sc.nextInt();

**for** (p= 0; p < m; p++)

**for** (q = 0; q < n; q++)

c[p][q] = a[p][jq] + b[p][q];

System.***out***.println("Sum of the matrices");

**for** (p= 0; p< m; p++)

{

**for** (q = 0; q< n; q++)

System.***out***.print(c[p][q] + "\t");

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

}

}

}

**Output:**

Enter no.of rows and cols

2

2

Enter the first matrix

1 2

3 4

Enter the second matrix

1 2

7 4

Sum of the matrices

2 4

10 8

**2.Write a java program for Matrix Multiplication?**

**import** java.util.Scanner;

**public class** MatrixMulti

{

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

{

**int** m, n, p, q;

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

System.***out***.println("Enter no.of rows and cols");

m = sc.nextInt();

n = sc.nextInt();

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

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

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

System.***out***.println("Enter the first matrix");

**for** (p = 0; p < m; p ++)

**for** (q= 0; q < n; q++)

a[p][ p] = sc.nextInt();

System.***out***.println("Enter the second matrix");

**for** (p = 0 ; p < m; p ++)

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

b[p][q] = sc.nextInt();

**for** (p = 0; p < m; p ++) {

**for** (q = 0; q < n; q++) {

**for**(**int** o=0;o<m;o++) {

c[p][q]+=a[p][o] \* b[o][q];

}

}

}

System.***out***.println("Sum of the matrices");

**for** (p = 0; p < m; p++)

{

**for** (q = 0; q < n; q++)

System.***out***.print(c[p][q] + "\t");

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

}

}

}

Output:

Enter no.of rows and cols

3

3

Enter the first matrix

1 1 1

2 2 2

3 3 3

Enter the second matrix

1 1 1

2 2 2

3 3 3

Sum of the matrices

6 6 6

12 12 12

18 18 18

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

**Class** Main{

**void** method(**int** p, **int** q) {

System.***out***.println("Multiplication of" +p+"and"+q+" is=" + (p \* q));

}

**void** method(**int** p, **int** q, **int** r) {

System.***out***.println("Multiplication of" +p+","+q+"and"+r+" is=" + (p \* q \* r));

}

}

**public** **class** Multiply {

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

Main m = **new** Main();

m.mul(4, 4);

m.mul(6, 9, 1);

}

}

**Output:**

Multiplication of4and4 is=16

Multiplication of6,9and1 is=54

**4. Write a java program to create a class Point with two data members x&y.Include all constructors and display().?**

**class** Demo{

**int** p;

**int** q;

Demo(int p,int q){

p = 30;

q = 40;

}

**public** **void** display(){

System.***out***.println("sum of"+p+" and "+q+" is: "+(p+q));

}

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

Demo d1 = **new** Demo();

d1.display();

}

}

**Output:**

sum of30 and 40 is:70

**5. Write a java program using static method?**

**class** Main{

**static** **int** rectangle(**int** l,**int** b){

**return** l\*b;

}

**static** **int** square(**int** s){

**return** s\*s;

}

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

**int** a=Main.*rectangle*(2,4);

**int** b=Main.*square*(3);

System.***out***.println("area of rectangle is: "+a);

System.***out***.println("area of square is: "+b);

}

}

**Output:**

area of rectangle is: 8

area of square is: 9

**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.

Conditional statements in java:

* if statement
* nested if statement
* if-else statement
* if-else-if statement
* Switch Case Statement

**2. Write the syntax of switch-case statement?**

**switch**(expression){

**case** value1:

  //code to be executed

**break**;

**case** value2:

  //code to be executed

**break**;

......

**default**:

  code to be executed if all cases are not matched;

}

**3. Write the difference between break and continue statement?**

The keywords break and continue keywords are part of control structures in Java. Sometimes break and continue seem to do the same thing but there is a difference between them.

The break keyword is used to breaks(stopping) a loop execution, which may be a for loop, while loop, do while or for each loop.

The continue keyword is used to skip the particular recursion only in a loop execution, which may be a for loop, while loop, do while or for each loop.

**4. What is looping statement?**

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.

There are three types of loops in Java.

* for loop
* while loop
* do-while loop

**5. Write the difference between while and do..while statement?**

Although Do While loop and While loop looks similar, they differ in the order of execution.

* In [**While loop**](https://www.tutorialgateway.org/java-while-loop/), the condition is tested at the beginning of the loop, and if the condition is True, then only statements in that loop will be executed. So, the While loop executes the code block only if the condition is True.
* In  [**Do While loop**](https://www.tutorialgateway.org/java-do-while-loop/), the condition is tested at the end of the loop. So, the Do While executes the statements in the code block at least once even if the condition Fails.

**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?**

In [object-oriented programming](https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP) , a class is a template definition of the method s and [variable](https://whatis.techtarget.com/definition/variable) s in a particular kind of [object](https://searchapparchitecture.techtarget.com/definition/object) . Thus, an object is a specific instance of a class; it contains real values instead of variables.

The class is one of the defining ideas of object-oriented programming. Among the important ideas about classes are:

* A class can have subclasses that can inherit all or some of the characteristics of the class. In relation to each subclass, the class becomes the superclass.
* Subclasses can also define their own methods and variables that are not part of their superclass.
* The structure of a class and its subclasses is called the class hierarchy.

**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 thatcreates an object using another object of the same Java class.

That'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?**

If a [class](https://www.javatpoint.com/object-and-class-in-java) has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

If we have to perform only one operation, having same name of the methods increases the readability of the [program](https://www.javatpoint.com/java-programs).

Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as a(int,int) for two parameters, and b(int,int,int) for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.

**12.What is Static variable?**

The **static keyword** in [Java](https://www.javatpoint.com/java-tutorial) is used for memory management mainly. We can apply static keyword with [variables](https://www.javatpoint.com/java-variables), methods, blocks and [nested classes](https://www.javatpoint.com/java-inner-class). The static keyword belongs to the class than an instance of the class.

The static can be:

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class

**13.What is access modifier?**

There are two types of modifiers in Java: **access modifiers** and **non-access modifiers**.

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

1. **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. **Default**: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.
3. **Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
4. **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

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

1. Instance method are methods which require an object of its class to be created before it can be called. Static methods are the methods in Java that can be called without creating an object of class.
2. Static method is declared with static keyword. Instance method is not with static keyword.
3. Static method means which will exist as a single copy for a class. But instance methods exist as multiple copies depending on the number of instances created for that class.
4. Static methods can be invoked by using class reference. Instance or non static methods are invoked by using object reference.
5. Static methods can’t access instance methods and instance variables directly. Instance method can access static variables and static methods directly.

**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