Assignment-2

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

**import** java.util.Scanner;

**public** **class** matrix\_addtion {

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

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

**int** row=sc.nextInt();

**int** col=sc.nextInt();

**int** a[][]= **new** **int**[row][col];

**int** b[][]=**new** **int**[row][col];

**int** c[][]=**new** **int**[row][col];

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

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

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

}

}

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

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

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

}

}

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

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

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

}

}

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

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

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

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

}

}

}

}

**output:**

2 2

1 1

2 2

1 1

2 2

2 2

4 4

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

**import** java.util.Scanner;

**public** **class** matrix\_multiplication {

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

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

**int** row=sc.nextInt();

**int** col=sc.nextInt();

**int** a[][]= **new** **int**[row][col];

**int** b[][]=**new** **int**[row][col];

**int** c[][]=**new** **int**[row][col];

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

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

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

}

}

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

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

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

}

}

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

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

c[i][j]=0;

**for**(**int** k=0;k<row;k++) {

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

}

}

}

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

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

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

}

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

}

}

}

output:

3 3

1 1 1

2 2 2

3 3 3

1 1 1

2 2 2

3 3 3

6 6 6

12 12 12

18 18 18

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

**package** anil;

**class** mo{

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

**int** c=a+b;

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

}

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

c=a\*b\*b\*c;

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

}

}

**public** **class** methodoverloading {

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

mo obj= **new** mo();

obj.method(4, 7);

obj.method(2, 3, 4);

}

}

output:

11

72

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

**class** hii{

**int** x;

**int** y;

hii(**int** x,**int** y){

**this**.x = x;

**this**.y =y;

}

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

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

}

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

hii d1 = **new** hii(1, 2);

d1.display();

}

}

output:

3

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

**class** Main{

**static** **int** add(**int** a,**int** b){

**return** a+b;

}

**static** **int** multi(**int** x,**int** y){

**return** x\*y;

}

}

**public** **class** static\_method {

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

**int** r1=Main.*add*(5,6);

**int** r2=Main.*multi*(2,3);

System.***out***.println(r1);

System.***out***.println(r2);

}

}

**output:**

11

6

**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?**

Looping statement are the statements execute one or more statement repeatedly several number of times. In java programming language there are three types of loops; while, for and do-while.

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?**

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'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 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?**

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