**Constructor in java**

In [Java](https://www.javatpoint.com/java-tutorial), a constructor is a block of codes similar to the method. It is called when an instance of the [class](https://www.javatpoint.com/object-and-class-in-java) is created. At the time of calling constructor, memory for the object is allocated in the memory.

It is a special type of method which is used to initialize the object.

**Rules for creating java constructor :**

There are basically two defined for the constructor

* Constructor name must be same as its class name
* Constructor must have no explicit return type

**Types of constructor in java**

There are two types of constructor in java :

* Default constructor(no – arg constructor)
* Parameterized constructor

**Default constructor(no – arg constructor) :**

* A constructor that have no parameter is known as default constructor .
* **Syntax of default constructor is :**

<class\_name>()

{

}

## **Example of default constructor**

|  |
| --- |
| In this example, we are creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation. |

**class** Student {

Student () {

System.out.println ("Welcome to KKWIEER");

}

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

Student s = **new** Student();

}

}

* Output:

Welcome to KKWIEER

**parameterized constructor :**

A constructor which has a specific number of parameters is called a parameterized constructor.

### **Why use the parameterized constructor?**

The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

### **Example of parameterized constructor**

In this example, we have created the constructor of Student class that have two parameters. We can have any number of parameters in the constructor .

**class** Student{

**int** id;

    String name;

Student(**int** i , String n){

    id = i ;

     name = n ;

    }

**void** display(){

System.out.println(id+" "+name);

}

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

    Student s1 = **new** Student(111,"Sneha");

    Student s2 = **new** Student(222,"Mayuri");

    s1.display();

    s2.display();

   }

}

Output:

111 Sneha

222 Mayuri

**enum :**

* enum was introduced in java 1.5
* An enum is a special type of data type which is a collection of constants .
* Actually enum is a just like a class that can have constant , method and private constructor .(If we don’t declare private , compiler internally creates constructor ) .
* enum constants are by default public , static and final that means it can not be changeable (can not be overriden) .
* Because of private constructor we cant create the instance of enum by using new keyword .
* enum can not extends other classes , but enum can implement interfaces .

|  |
| --- |
| enum Month  {  JAN , FEB , MAR ;  } |

Internally array arrange their content in an array .

|  |  |  |
| --- | --- | --- |
| JAN | FEB | MAR |

0 1 2

So we can print the content of enum by using loop (for each loop)

We can also print the index of content of enum .

**Example :**

public class ABC{

public enum Month{

JAN(1) , FEB(2) , MAR(3);

private int value ;

private Month(int value){

this.values = value ;

}

}

public static void main(String args[]){

for(Month m : Month.values())

System.out.println(m + “ “ + m.value) ;

}

}

**Important Note –** Must have semicolon at the end when there is a enum field or method .