Java Constructors

What is Constructor in Java?

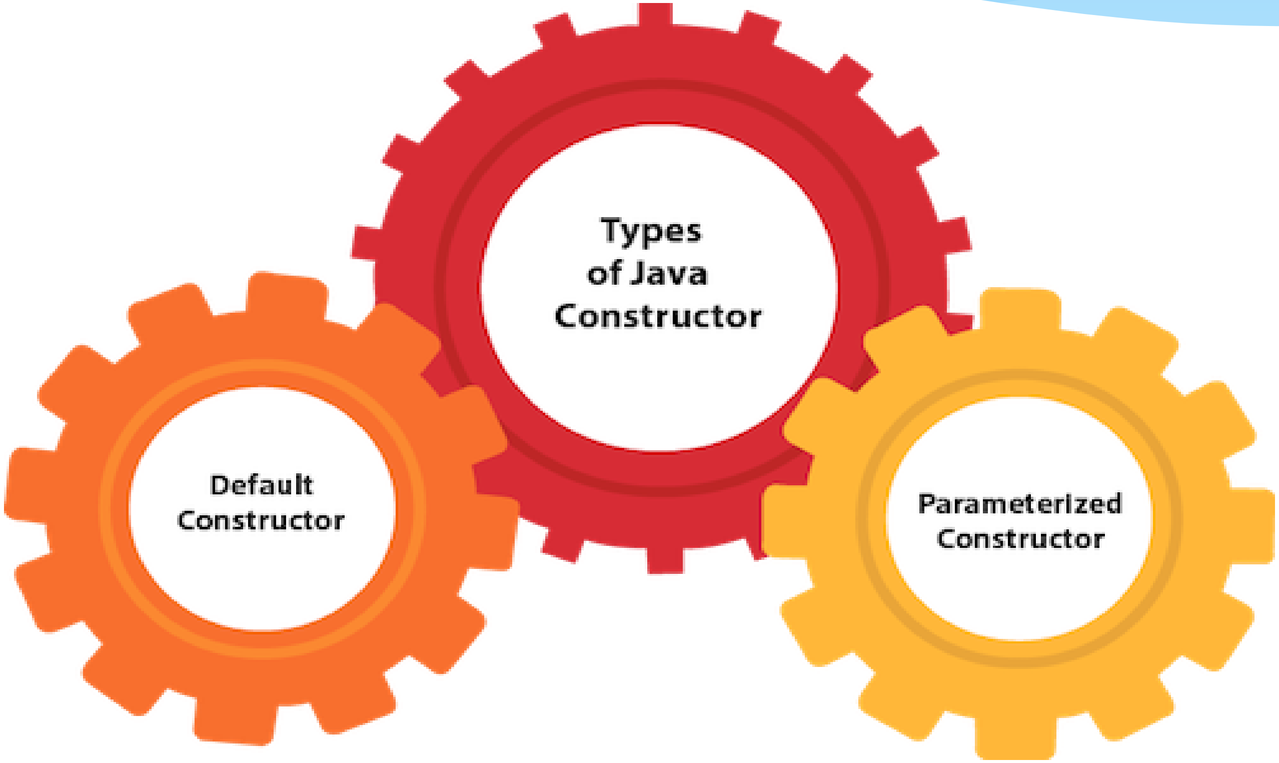
- The constructor is a special type of method in Java that doesn’t perform any action, rather than initializing values to the variables.
- Constructors in Java are special types of methods that are used to initialize the objects of the class. Constructors are called at the time of object creation of the class.

Rules for creating Java constructor

There are two rules defined for the constructor.

- Constructor name must be the same as its class name
- A Constructor must have no explicit return type
- A Java constructor cannot be abstract, static, final, and synchronized

Types of Java constructors



Java Default Constructor

- A constructor is called "Default Constructor" when it doesn't have any parameter.
- Syntax of default constructor:
- <class_name>(){}

Difference between constructor and method in Java

Java Constructor	Java Method
A constructor is used to initialize the state of an object.	A method is used to expose the behavior of an object.
A constructor must not have a return type.	A method must have a return type.
The constructor is invoked implicitly.	The method is invoked explicitly.
The Java compiler provides a default constructor if you don't have any constructor in a class.	The method is not provided by the compiler in any case.
The constructor name must be same as the class name.	The method name may or may not be same as the class name.

Example: Default Constructor

```
import java.io.*;

class GFG {
   GFG() { System.out.println("Default constructor"); }
   public static void main(String[] args)
   {

      GFG hello = new GFG();
   }
}
```

- **Output:** Default constructor

Java Parameterized Constructor

- A constructor which has a specific number of parameters is called a parameterized constructor.
- The parameterized constructor is used to provide different values to distinct objects

Example of Parameterized Constructor

```
/Java Program to demonstrate the use of the parameterized constructor.
class Student4{
    int id;
    String name;
    //creating a parameterized constructor
    Student4(int i,String n){
    id = i;
    name = n;
    }
    //method to display the values
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
    //creating objects and passing values
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    //calling method to display the values of object
    s1.display();
    s2.display();
   }
}
```

OUTPUT

```
111 Karan
 222 Aryan
```

Constructor Overloading in Java

- In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.
- Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

Example of Constructor Overloading

```
//Java program to overload constructors
class Student5{
  int id;
  String name;
  int age;
  //creating two arg constructor
  Student5(int i,String n){
  id = i;
  name = n;
  }
  //creating three arg constructor
  Student5(int i,String n,int a){
  id = i;
  name = n;
  age=a;
  }
  void display(){System.out.println(id+" "+name+" "+age);}

  public static void main(String args[]){
  Student5 s1 = new Student5(111,"Karan");
  Student5 s2 = new Student5(222,"Aryan",25);
  s1.display();
  s2.display();
 }
}
```

Output:

```
111 Karan 0 222
Aryan 25
```