Constructor in Java

**Constructor in java** is a special type of method that is used to initialize the object.

Java constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.

### Rules for creating java constructor

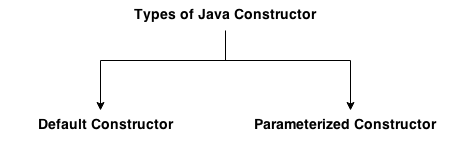
There are basically two rules defined for the constructor.

1. Constructor name must be same as its class name
2. Constructor must have no explicit return type

### Types of java constructors

There are two types of constructors:

1. Default constructor (no-arg constructor)
2. Parameterized constructor



## Default Constructor

|  |
| --- |
| A constructor that has no parameter is known as default constructor. |
| Syntax of default constructor:  1. <class\_name>(){}  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. |  1. class Bike1 2. { 3. Bike1(){System.out.println("Bike is created"); 4. } 5. public static void main(String args[]){ 6. Bike1 b=new Bike1(); 7. } 8. } |
|  |
| Default constructor that displays the default values  1. **class** Student3 2. { 3. **int** id; 4. String name; 6. **void** display() 7. { 8. System.out.println(id+" "+name); 9. } 11. **public** **static** **void** main(String args[]) 12. { 13. Student3 s1=**new** Student3(); 14. Student3 s2=**new** Student3(); 15. s1.display(); 16. s2.display(); 17. } 18. }  Parameterized constructor  |  | | --- | | A constructor that has parameters is known as parameterized constructor. | | Why use parameterized constructor?  |  | | --- | | Parameterized constructor is used to provide different values to the distinct objects. |   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. |  1. class Student4{ 2. int id; 3. String name; 5. Student4(int i,String n){ 6. id = i; 7. name = n; 8. } 9. void display(){System.out.println(id+" "+name);} 11. public static void main(String args[]){ 12. Student4 s1 = new Student4(111,"Ahmed"); 13. Student4 s2 = new Student4(222,"Ali"); 14. s1.display(); 15. s2.display(); 16. } 17. }  Constructor Overloading in Java  |  | | --- | | Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists.The compiler differentiates these constructors by taking into account the number of parameters in the list and their type. |  Example of Constructor Overloading  1. **class** Student5{ 2. **int** id; 3. String name; 4. **int** age; 5. Student5(**int** i,String n){ 6. id = i; 7. name = n; 8. } 9. Student5(**int** i,String n,**int** a){ 10. id = i; 11. name = n; 12. age=a; 13. } 14. **void** display(){System.out.println(id+" "+name+" "+age);} 16. **public** **static** **void** main(String args[]){ 17. Student5 s1 = **new** Student5(111,"Ahmed"); 18. Student5 s2 = **new** Student5(222,"Ali",25); 19. s1.display(); 20. s2.display(); 21. } 22. }  Difference between constructor and method in java There are many differences between constructors and methods. They are given below.   |  |  | | --- | --- | | **Java Constructor** | **Java Method** | | Constructor is used to initialize the state of an object. | Method is used to expose behavior of an object. | | Constructor must not have return type. | Method must have return type. | | Constructor is invoked implicitly. | Method is invoked explicitly. | | The java compiler provides a default constructor if you don't have any constructor. | Method is not provided by compiler in any case. | | Constructor name must be same as the class name. | Method name may or may not be same as class name. | | |