Constructors in Java

Constructors are used to initialize the object’s state. Like [methods](https://www.geeksforgeeks.org/methods-in-java/), a constructor also contains **collection of statements(i.e. instructions)** that are executed at time of Object creation.

**Need of Constructor**  
Think of a Box. If we talk about a box class then it will have some class variables (say length, breadth, and height). But when it comes to creating its object(i.e Box will now exist in computer’s memory), then can a box be there with no value defined for its dimensions. The answer is no.  
So constructors are used to assign values to the class variables at the time of object creation, either explicitly done by the programmer or by Java itself (default constructor).

**When is a Constructor called ?**  
Each time an object is created using **new()** keyword at least one constructor (it could be default constructor) is invoked to assign initial values to the **data members**of the same class.

* Constructor(s) of a class must has same name as the class name in which it resides.
* A constructor in Java can not be abstract, final, static and Synchronized.
* Access modifiers can be used in constructor declaration to control its access i.e which other class can call the constructor.
* Constructor should not have return type

**Types of constructor**

There are two type of constructor in Java:

1. **No-argument constructor:**A constructor that has no parameter is known as default constructor. If we don’t define a constructor in a class, then compiler creates **default constructor(with no arguments)** for the class. And if we write a constructor with arguments or no-arguments then the compiler does not create a default constructor.  
   Default constructor provides the default values to the object like 0, null, etc. depending on the type.

**Parameterized Constructor:**A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with your own values, then use a parameterized constructor.

class Geek

{

    // data members of the class.

    String name;

    int id;

    // constructor would initialize data members

    // with the values of passed arguments while

    // object of that class created.

    Geek(String name, int id)

    {

        this.name = name;

        this.id = id;

    }

}

**Constructor Overloading**

Like methods, we can overload constructors for creating objects in different ways. Compiler differentiates constructors on the basis of numbers of parameters, types of the parameters and order of the parameters.

**How constructors are different from methods in Java?**

* Constructor(s) must have the same name as the class within which it defined while it is not necessary for the method in java.
* Constructor(s) do not return any type while method(s) have the return type or **void** if does not return any value.
* Constructor is called only once at the time of Object creation while method(s) can be called any numbers of time.

# Constructor Chaining

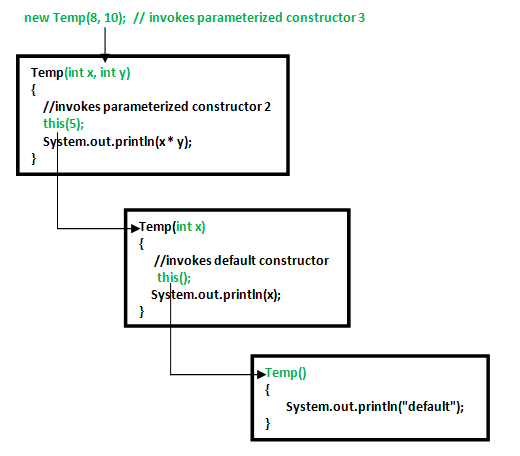
Constructor chaining is the process of calling one constructor from another constructor with respect to current object.  
Constructor chaining can be done in two ways:

* **Within same class**: It can be done using **this()** keyword for constructors in same class
* **From base class:**by using **super()** keyword to call constructor from the base class.

Constructor chaining occurs through **inheritance**. A sub class constructor’s task is to call super class’s constructor first. This ensures that creation of sub class’s object starts with the initialization of the data members of the super class. There could be any numbers of classes in inheritance chain. Every constructor calls up the chain till class at the top is reached.

**Why do we need constructor chaining ?**  
This process is used when we want to perform multiple tasks in a single constructor rather than creating a code for each task in a single constructor we create a separate constructor for each task and make their chain which makes the program more readable.

**Constructor Chaining within same class using this() keyword :**

[](https://media.geeksforgeeks.org/wp-content/uploads/Constructor-Chaining-In-Java1.png)

**Rules of constructor chaining :**

1. The **this()** expression should always be the first line of the constructor.
2. There should be at-least be one constructor without the this() keyword (constructor 3 in above example).
3. Constructor chaining can be achieved in any order.

Note : Similar to constructor chaining in same class, **super()** should be the first line of the constructor as super class’s constructor are invoked before the sub class’s constructor.

**Alternative method : using Init block**:  
When we want certain common resources to be executed with every constructor we can put the code in the [**init block**](https://www.geeksforgeeks.org/g-fact-26-the-initializer-block-in-java/). Init block is always executed before any constructor, whenever a constructor is used for creating a new object.