This and Super keyword:

Constructor:

It can be used for object creation and initialization .

Objects can be initaialized at the time of object creation itself

1.name shd be same of a class name

2.it is similar to function but it does not return anything,no void,no return type,no static

1) Default constructor:

With no i/p parameters

2)Parameterized constructor:

With i/p parameters

Can we overload a constructor?

Constructor can also be overloaded similar to methods

How to call a constructor?

We shd call a constructor inside main method .

We need to create an object and need not call it using object reference.

Once ,we create an object, in main method we can directly call the constructor.

Note:

While creating the class,a default constructor will be created automatically,and it will be hidden.

Constructor definition:

1.It is a class entity which is used to define some class features while creating the object in the form of global variables.

2.It looks lik func,but it does not return any value.name shd be same of class name.(no void,no return type,no static)

\

How to call the constructor?

It cannot be calles lik a method where a method is called by object reference of that class.

So,it is called immediately,when we create object of the class.

Whenever an object of a particular class is created,immediately default const will be called even if we have not created any bcz.default const is available hiiden with the class.

To initaiaize global variables.we use this keyword

This.global variable=local variable

Refer:

A constructor and a method are two different things. The fact that you can write the same or similar code inside them is irrelevant.

When a new object is created a constructor is called. If you don't s

Now since a constructor is always being called, you have an option here: do you let the default constructor execute or do you create one yourself? Perhaps there are fields that need to be initialized in a way other than their default values. Or perhaps you need to not allow creating an object without providing some values. If you define a constructor yourself, the compiler *does not create a default one for you*.

So if I have public Person(String firstName, String lastName) {} then I have created a specific rule that is again enforced by the system: a new object of class Person cannot be created unless you give a first name and last name:

Person p = new Person(); // this would give a compile error

Person p = new Person("John", "Smith"); // this is the only way to create an object now

Using a method you cannot enforce this. The programmer using your class might call your method or not. The constructor is a part of the *lifecycle* of the object. Methods define the *behaviour* of the object

pecify one the compiler will create a default one for you. This is the place where initializaton of the object's fields takes place and memory is allocated for the object. This is a concept that all object-oriented languages have. A new object must be initialized somehow. Memory needs to be allocated. In Java you don't manage the memory yourself (at least not directly anyway) so this is the purpose of the constructor. Note that since a constructor is always executed, this behaviour is **enforced** as soon as you call e.g. Person p = new Person();.

**Generally:**  
A constructor is **always** called once when you create a new Object of this class, and you **can´t** call it manually.  
And **don´t** do "real" work in a constructor, as it will slow down the creation of objects of this class - only initialize your class/members there.

Super keyword:

Used to call parent class parametereized constructor.

Shd be written within a child class constructor always.

It shd be written as a first statement inside the constructor.

Note:even if we don’t use super keyword,default parent class const will be called.

We cannot create constructor within methods.

**Ctrl+Shift+F -🡪to format the code**