

Constructors - 2:

The 1st line inside every constructor should be either `super()` or `this()`. If we don't write anything compiler will always generate `super()`.

Various ways to call the constructor:

- a) `Test test = new Test();`
- b) `new Test();`
- c) `this();`
- d) `super();`

Note:

`this()` and `super()` can be used in constructors only. If we write it at any other place, it will be compile time error.

Constructor doesn't inherit and hence constructor overriding is not possible. Constructors can only be overloaded.

this() -

`this()` is used to call the current class constructor.

Program-1.1 to use of current class constructor-

```
package com.test;
```

```
class Test {
```

```
    public Test() {  
        System.out.println("Test constructor called using" + " this  
keyword..");  
    }
```

```
    public Test(int x) {  
        this(); // invoke the constructor of current class  
    }
```

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

```
        Test test = new Test(10);
```

```
}  
  
}
```

Output-

Test constructor called using this keyword..

Program-1.2 to use of current class constructor-

```
public class Example {  
  
    Example() {  
        this("Java");  
        System.out.println("Inside Constructor without parameter");  
    }  
  
    Example(String str) {  
        System.out.println("Inside Constructor with parameter" + str);  
    }  
  
    public static void main(String[] args) {  
        Example obj = new Example();  
    }  
}
```

Output-

Inside Constructor with String parameter as Java
Inside Constructor without parameter

super() –

It is used to call the immediate parent class constructor

Program-3 to use of immediate super class constructor-

```
class Parent {  
    Parent() {  
        System.out.println("Parent class constructor.");  
    }  
}  
  
class Child extends Parent {  
    Child() {  
        super();  
    }  
}  
  
public class TestMain {  
    public static void main(String[] args) {  
        Child c= new Child();  
    }  
}
```

Output-

Parent class constructor.

Note :

- 1) If we want to write super() or this() in a constructor then it has to be first line of constructor. If we are writing it anywhere else then it will be compile time error.
- 2) We can't use super() or this() at the same time in same constructor. We can either use super() or this().
- 3) If we don't write any super() or this(), then compiler will always place super() by default in first line of constructor.

Summary :

Programmer Code	Compiler Code
<pre>class Demo{ }</pre>	<pre>class Demo{ Demo(){ super(); } }</pre>
<pre>public class Demo{ }</pre>	<pre>public class Demo{ public Demo(){ super(); } }</pre>
<pre>class Demo{ void Demo(){ } }</pre>	<pre>class Demo{ void Demo(){ } Demo(){ super(); } }</pre>
<pre>class Demo{ Demo(int i){ } }</pre>	<pre>class Demo{ Demo(int i){ super(); } }</pre>
<pre>class Demo{ Demo(int i){ super(); } }</pre>	<pre>class Demo{ Demo(int i){ super(); } }</pre>
<pre>class Demo{ Demo(int i){ this(); } }</pre>	<pre>class Demo{ Demo(int i){ this(); } }</pre>

<pre>} Demo(){ } }</pre>	<pre>} Demo(){ Super(); } }</pre>
---	---