DEFAULT CONSTRUCTOR

1. Is always no-arg constructor

2. The access modifier of default constuctor is exactly same as class(This rule is applicable for public and default)

3. It contains only one line

***super();***

Is a no argument call to super class constructor.

|  |  |
| --- | --- |
| Programmer’s code | Compiler Generate Code |
| Class Test{} | **class** Base {  Base(){ **super**(); }  } |
| Public class Test {} | **Public class** Base {  public Base(){ **super**(); }  } |
| Public class Test {  void Test() {}  } | **Public class** Base {  public Base(){ **super**(); }  void Test() {}  } |
| Class Test {  Test() {}  } | Class Test {  Test() {  super();  }  } |
| Class Test{  Test(int i) {  super();  }  } | Class Test{  Test(int i) {  super();  }  }//same |
| Class Test{  Test(int i) {  this(10);  }  Test(int i){}  } | Class Test{  Test(int i) {  this(10);  }  Test(int i){ super();}  } |

Case 1:

1. Call to the **super or this** should on the first line of the constructor

2. We can take **either super or this but not both**

3. We can use **super or this** only inside constructor, if we trying to use outside of constructor we will get compile-time error.

|  |  |
| --- | --- |
| Super(), this() | Super, this |
| These are constructor calls | These are keywords to refer super and current class instance members. |
| Only inside constructor | Everywhere expect static area |
| We can use only once in constructor | We can use any number of times |