

Rules of the Constructors in the Inheritance

- The default constructor of the super class will be invoked by the default constructor of the sub implicitly
- If the Super Class is defined with both default and parameterized constructor, In this is case also the default constructor of super class will be invoked by the default constructor of the Subclass implicitly. If you want invoke the parameterized constructor of the super class by the default constructor of subclass then we have make to use of “super([argument])”
- If the Super class and Sub class both are defined with the parameterized constructors ,then we must call the parameterized constructor of the super class by the parameterized constructor of subclass explicitly. Otherwise parameterized constructor of the sub class is trying to call default constructor of the super as it is not existed we will be getting an Error.

Example :

```
class SuperA  
{  
    SuperA()
```

```
        { System.out.println("Def const of SA "); }  
    }  
    class SubB extends SuperA  
    {  
        SubB()  
        { System.out.println("Def const of SB "); }  
    }  
    class R1  
    {  
        public static void main(String args[ ])  
        { SubB sb=new SubB(); }  
    }
```

Example 2:

```
class SuperA  
{  
    SuperA()  
    { System.out.println("Def const of SA "); }  
  
    SuperA(int x)  
    { this();  
      System.out.println("Para const of SA : "+x); }  
}  
class SubB extends SuperA  
{  
    SubB()  
    {  
        super(123);  
        System.out.println("Def const of SB ");  
    }  
}
```

```
class R2
{
    public static void main(String args[ ])
    { SubB sb=new SubB(); }
}
```

Note:

➤ **this() vs super()**

- if you want call an existed constructor inside of another constructor of the same class then we have to make user of "this()"
- If you want call an existed constructor of the super class into another constructor of the sub class then we have to make use of "super()"

Example 3:

class SuperA

```
{
    SuperA(int x)
    { System.out.println("Para const of SA "+x); }
}
```

class SubB extends SuperA

```
{
    SubB(int x)
    { super(145); //try by removing this stmt
      System.out.println("Para const of SB : "+x); }
}
```

class R3

```
{  
    public static void main(String args[ ])  
    {  
        SubB sb=new SubB(123);  
    }  
}
```

Note :

- If you call either default or parameterized constructor of the subclass then internally JVM will call “Default constructor “ of the Super class only.