Constructor in Inheritance

- In the inheritance, the constructors never get inherited to any child class.
- In java, the default constructor of a parent class called automatically by the constructor of its child class. That means when we create an object of the child class, the parent class constructor executed, followed by the child class constructor executed.
- if the parent class contains both default and parameterized constructor, then only the default constructor called automatically by the child class constructor. Parameterized constructors are called explicitly by child class through **super() or parent class attributes are initialized by the child class constructor.**

```
class P{
	int a;
	P(){
		System.out.println("Inside ParentClass constructor!");
	}
}
class C extends {

	C(){
		System.out.println("Inside ChildClass constructor!!");
	}
}
class CC extends C{

	CC(){
```

```
                System.out.println("Inside ChildChildClass constructor!!");
        }
}
public class test {

        public static void main(String[] args) {

                CC obj = new CC();

        }

}
```

o/p

Inside ParentClass constructor!

Inside ChildClass constructor!!

Inside ChildChildClass constructor!!

How to call parameterized constructor of parent class in inheritance

```
class p
{
 int x;
 p()
{
 System.out.println("super constructor");
```

```
}
//parameterized constructor of parent
p(int x)
{
 this.x=x;
 System.out.println("p class parameterized constructor");

}
}

class c extends p
{
 int y;
 c()
 {
  System.out.println("sub constructor");
 }

 c(int b)
 {

  super(56); // call parameterized constructor of parent class

  y=b;
 }
```

```
}


class consinherit
{
 public static void main(String[] s)
 {
   c cob=new c();
   c cob1=new c(12);// c parameterized constructor is called
   System.out.println(cob1.x+"   "+cob1.y);


}
}
```