Inheritance and constructors in Java

Difficulty Level : Easy   Last Updated : 23 Apr, 2021

In Java, constructor of base class with no argument gets automatically called in derived class constructor. For example, output of following program is:

Base Class Constructor Called

Derived Class Constructor Called

```java
// filename: Main.java
class Base {
  Base() {
    System.out.println("Base Class Constructor Called ");
  }
}

class Derived extends Base {
  Derived() {
    System.out.println("Derived Class Constructor Called ");
  }
}

public class Main {
  public static void main(String[] args) {
    Derived d = new Derived();
  }
}
```

But, if we want to call parameterized constructor of base class, then we can call it using super(). The point to note is **base class constructor call must be the first line in derived class constructor**. For example, in the following program, super(_x) is first line derived class constructor.

```java
// filename: Main.java
class Base {
  int x;
  Base(int _x) {
    x = _x;
  }
}

class Derived extends Base {
  int y;
  Derived(int _x, int _y) {
    super(_x);
    y = _y;
  }
  void Display() {
    System.out.println("x = "+x+", y = "+y);
  }
}

public class Main {
  public static void main(String[] args) {
    Derived d = new Derived(10, 20);
    d.Display();
  }
}
```

Output:

```
x = 10, y = 20
```

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.