Constructor Overloading

Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists. The compiler differentiates these constructors by taking the number of parameters, and their type.

In other words whenever same constructor is existing multiple times in the same class with different number of parameters or order of parameters or type of parameters is known as Constructor overloading.

In general constructor overloading can be used to initialize same or different objects with different values.

Syntax

```
class  ClassName
{
ClassName()
{
.....
.....
}
ClassName(datatype1 value1)
{.....}
ClassName(datatype1 value1, datatype2 value2)
{.....}
ClassName(datatype2 variable2)
{.....}
ClassName(datatype2 value2, datatype1 value1)
{.....}
.....
}
```

Why overriding is not possible at constructor level.

The scope of constructor is within the class so that it is not possible to achieved overriding at constructor level.

Example of default constructor, parameterized constructor and overloaded constructor

Example

```
class Test
{
int a, b;
Test ()
{

System.out.println("I am from default Constructor...");
a=1;
b=2;
System.out.println("Value of a ="+a);
System.out.println("Value of b ="+b);
}
Test (int x, int y)
{
System.out.println("I am from double Parameterized Constructor");
a=x;
b=y;
System.out.println("Value of a ="+a);
```

```
System.out.println("Value of b ="+b);
}
Test (int x)
{
System.out.println("I am from single Parameterized Constructor");
a=x;
b=x;
System.out.println("Value of a ="+a);
System.out.println("Value of b ="+b);
}

Test (Test T)
{
System.out.println("I am from Object Parameterized Constructor...");
a=T.a;
b=T.b;
System.out.println("Value of a ="+a);
System.out.println("Value of b ="+b);
}
}
class TestDemo2
{
public static void main (String k [])
{
Test t1=new Test ();
Test t2=new Test (10, 20);
Test t3=new Test (1000);
Test t4=new Test (t1);
}
}
```

Note By default the parameter passing mechanism is call by reference.