**Constructor:**

Constructor is object initializer.

Same name as class.

NO return type.

NOT a function/method.

Constructor types:

1. Default Constructor

2. No-arg constructor

3. Parameterized constructor

1. Default Constructor - We do not write the

constructor, but mention it while creating an

object.

2. No-arg constructor - We write a constructor ourselves

Looks like a method, but not a method and has no return type.

Also, the name is exactly the same as the Class it is in.

3. Parametrized Constructor - We write a constructor ourselves

and pass a parameter or two. It also looks like a method,

except it has no return type and the name is same as the class it

is in.

public class Constructor {

public static void main(String[] args) {

Class1 obj1 = new Class1(); // Default Constructor

obj1.iVar1 = 90;

System.out.println(obj1.iVar1);

Class2 obj2 = new Class2(); // No argument Constructor

System.out.println(obj2.iVar2);

Class2 obj3 = new Class2(112,115); // Parametrized Constructor

System.out.println(obj3.iVar2+" "+obj3.iVar3+" "+obj3.iVar4);

}

}

class Class1 {

int iVar1;

}

Output:

90

45

0 112 115

**Super Keyword:**

Whenever a child class constructor gets invoked it implicitly invokes the constructor of parent class.

You can also say that the compiler inserts a super(); statement at the beginning of child class constructor.

class ParentClass{

int iVar1;

ParentClass(int a){

iVar1 = a;

}

}

class ChildClass extends ParentClass{

int iVar2, iVar3;

ChildClass(int a, int b, int c){

super(a);

iVar2 = b;

iVar3 = c;

}

}

public class SuperKeyword {

public static void main(String[] args) {

ChildClass obj1 = new ChildClass(1,2,3);

System.out.println(obj1.iVar1+" "+ obj1.iVar2+" "+obj1.iVar3);

}

}

Output:

1 2 3

**Copy Constructor:**

A copy constructor is used for copying the values of one object to another object.

class RandomClass{

int iVar1, iVar2;

RandomClass(int a, int b){

iVar1 = a;

iVar2 = b;

}

RandomClass(RandomClass obj){

iVar1 = obj.iVar1;

iVar2 = obj.iVar2;

}

}

public class CopyConstructor {

public static void main(String[] args) {

RandomClass obj1 = new RandomClass(113,114);

RandomClass obj2 = new RandomClass(obj1);

System.out.println(obj2.iVar1+" "+obj2.iVar2);

}

}

Output:

113 114