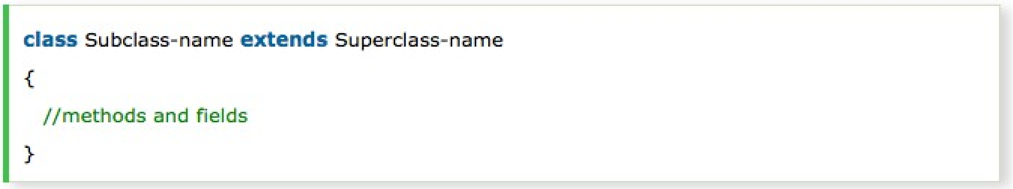
# Inheritance:-

The process of getting properties and behaviours from one class to another class is called inheritance.

**Properties** : variables
**Behaviours :** methods

1. The main purpose of the inheritance is code extensibility whenever we are extending automatically the code is reused.
2. In inheritance one class giving the properties and behaviour and another class is taking the properties and behaviour.
3. Inheritance is also known as is-a relationship means two classes are belongs to the same hierarchy.
4. By using extends keyword we are achieving inheritance concept.
5. In the inheritance the person who is giving the properties is called parent the person who is taking the properties is called child.
6. To reduce length of the code and redundancy of the code sun peoples introducing inheritance concept.

### Syntax of Inheritance

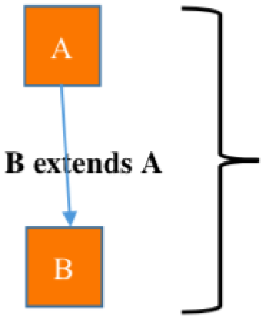


```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

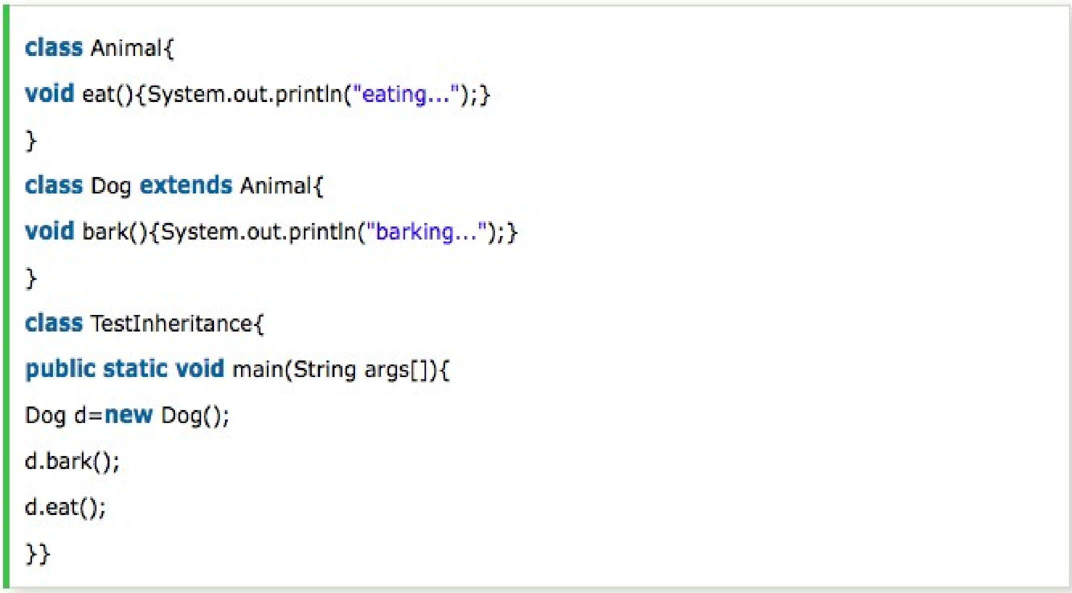
## Types of inheritance:-

### Single inheritance:-



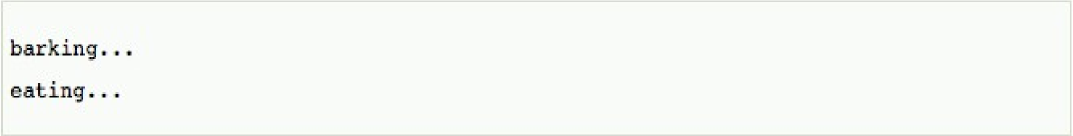
Extending the properties from single parent class to single child class is called single inheritance.
Class A is Parent class . Class B is Child class

**Ex.**



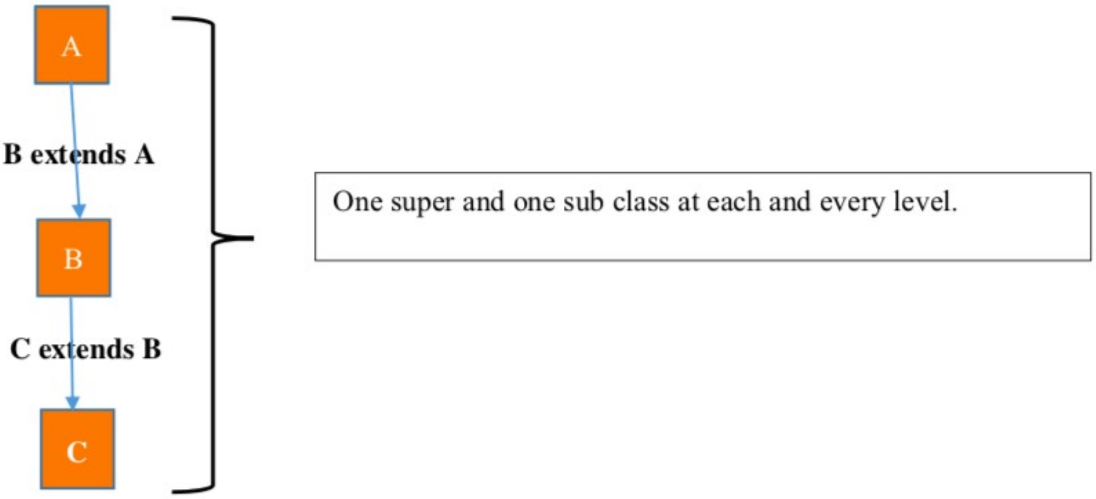
```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
}}
```

Output:



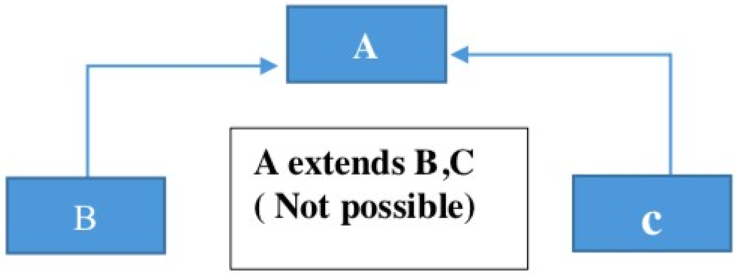
```
barking...
eating...
```

## Multilevel inheritance:-



One super and one sub class at each and every level.

## Multiple inheritance:-

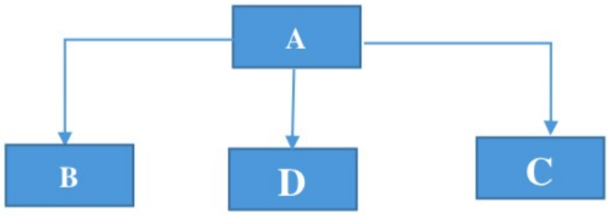


The process of getting properties and behaviours form more than one super class to the one child class. The multiple inheritance is not possible in the java language so one class can extends only one class at time it is not possible to extends more than one class at time.

**Class A extends B ----> possible**
**Class A extends C ----> possible**
**Class A extends C,B ----> not possible**

## Hierarchical inheritance:-

The process of getting properties and behaviours from one super class to the more than one sub classes is called hierarchical inheritance.



## Hybrid inheritance:-

Combination of any two inheritances is called as hybrid inheritance. If are taking the multilevel and hierarchical that combination is called hybrid inheritance.

```
//Before inheritance
class A
{
Void m1();
Void m2();
Void m3();
}

// Two new methods added m4() and m5() to
class A
class B
{
Void m1();
Void m2();
Void m3();
Void m4();
Void m5();
}
// one new method  added m6()
class C
{
Void m1();
Void m2();
Void m3();
Void m4();
Void m5();
Void m6();
}
```

```
//After inheritance

class A
{
Void m1();
Void m2();
Void m3() ;
}

class B extends A {

Void m4();
Void m5();
}

Class C extends B {
Void m6();
}
```

**Note 1:-**

It is possible to create objects for both parent and child classes.

1. If we are creating object for parent class it is possible to call only parent specific methods.

```
A a=new A();
a.m1();
a.m2();
a.m3();
```

2. if we are creating object for child class it is possible to call parent specific and child specific.

```
B b=new B();
b.m1();
b.m2();
b.m3();
b.m4();
b.m5();
```

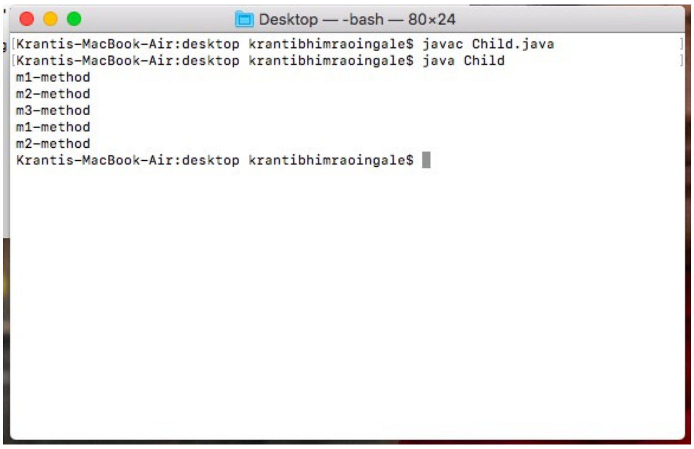
3. if we are creating object for child class it is possible to call parent specific methods and child specific methods.

```
C c=new C();
 c.m1();
 c.m2();
 c.m3();
 c.m4();
 c.m5();
 c.m6();
```

Ex:-

```
class Parent
{
	public void m1()
	{
	System.out.println("m1-method");
	}
	public void m2()
	{
	System.out.println("m2-method");
	}
}
class Child extends Parent
{
public void m3()
	{
	System.out.println("m3-method");
	}
	public static void main(String[] args)
	{
	Child c=new Child();
	c.m1();
	c.m2();
	c.m3();
	Parent p=new Parent();
	p.m1();
	p.m2();
	}
}
```

Output:



```
Krantis-MacBook-Air:desktop krantibhimraoingale$ javac Child.java
Krantis-MacBook-Air:desktop krantibhimraoingale$ java Child
m1-method
m2-method
m3-method
m1-method
m2-method
Krantis-MacBook-Air:desktop krantibhimraoingale$
```

**Note:-**

1. Every class in the java programming is a child class of Object.
2. The root class for all java classes is Object class.
3. The default package in the java programming is java.lang package.