**Inheritance**

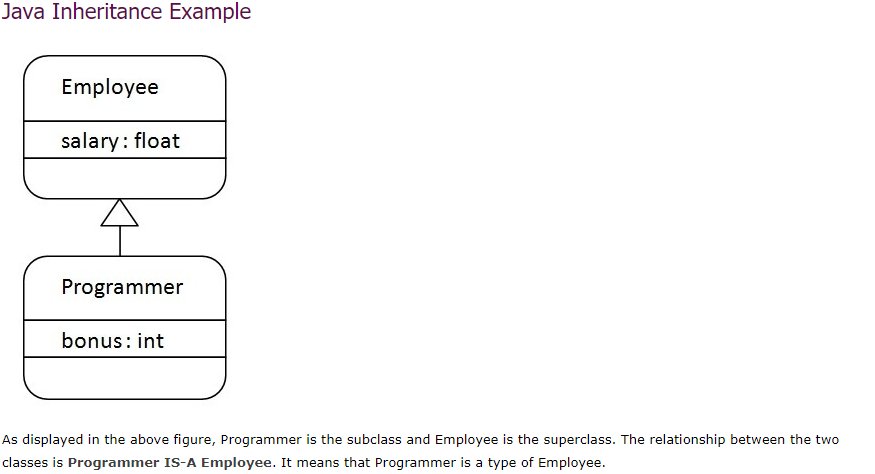
Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

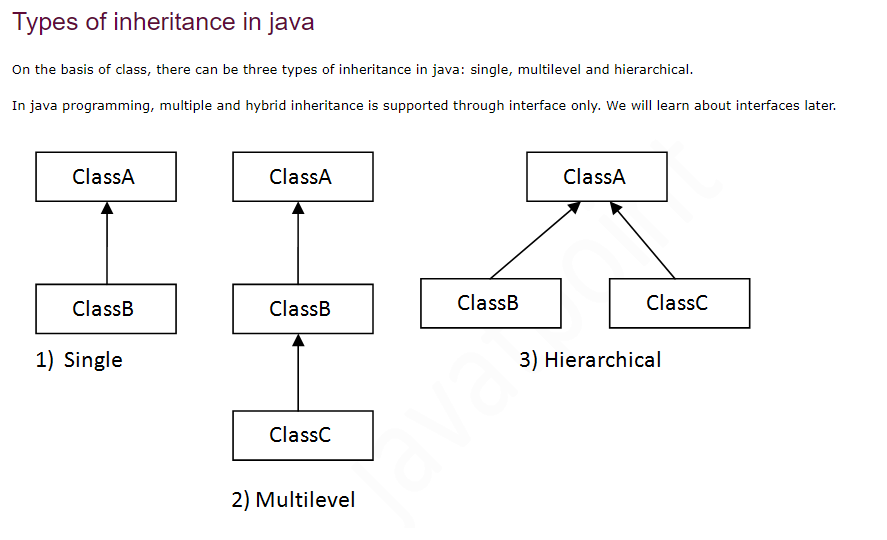
The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

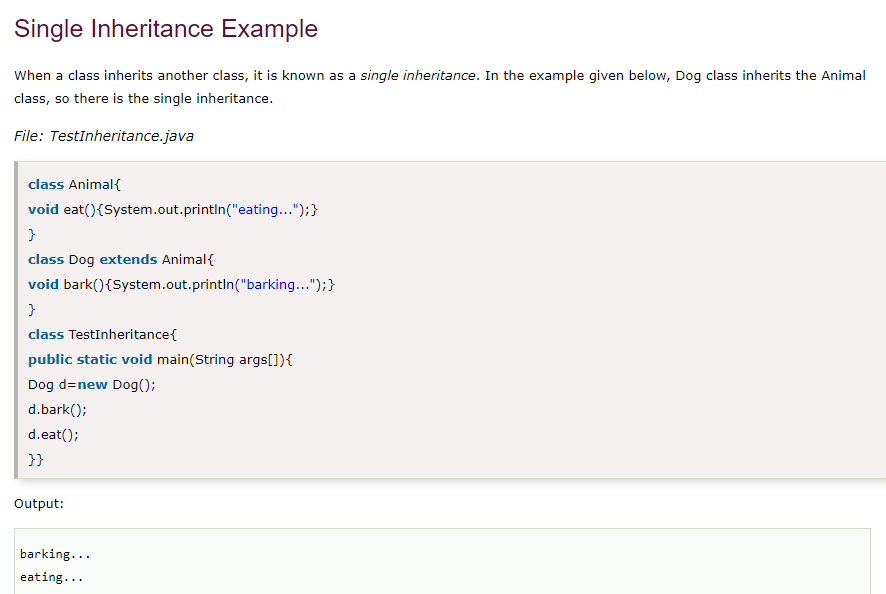
Inheritance represents the **IS-A relationship**

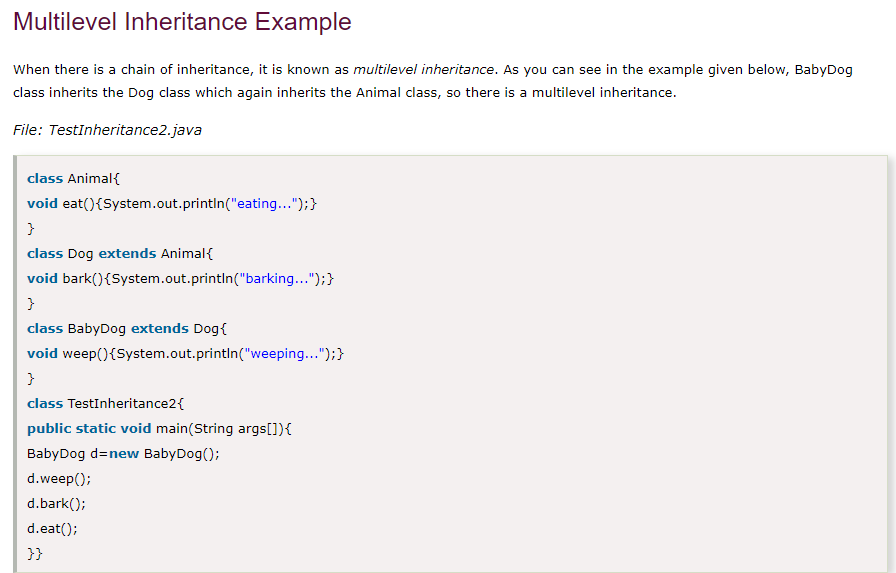
Why use inheritance in java

* For [Method Overriding](https://www.javatpoint.com/method-overriding-in-java) (so [runtime polymorphism](https://www.javatpoint.com/runtime-polymorphism-in-java) can be achieved).
* For Code Reusability.

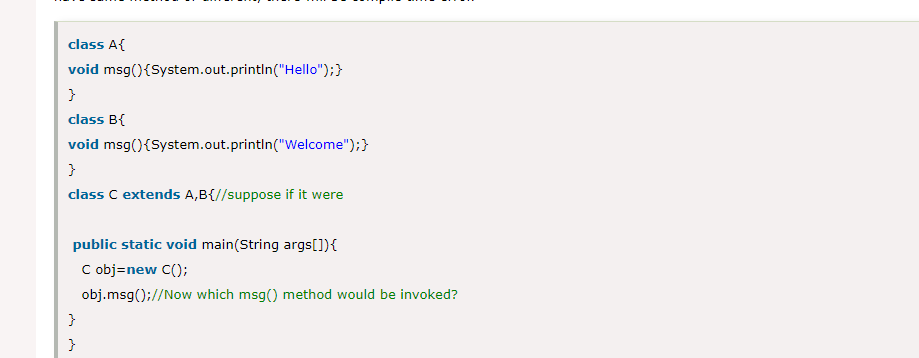




****

****

## Q) Why multiple inheritance is not supported in java?

****

**Code:**

class Animal {

public void displayInfo() {

System.out.println("I am an animal.");

}

}

class Dog extends Animal {

@Override

public void displayInfo() {

System.out.println("I am a dog.");

}

}

class Main {

public static void main(String[] args) {

Dog d1 = new Dog();

d1.displayInfo();

}

}

Super keyword:

The **super** keyword in Java is a reference variable which is used to refer immediate parent class object.

Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

## Usage of Java super Keyword

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

## 1) super is used to refer immediate parent class instance variable.