

Sub class/child class/ derived class

super/parent/base class

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of **OOPs** (Object Oriented programming system).

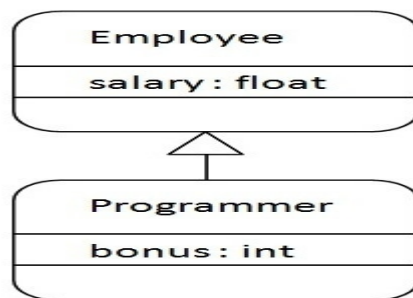
The idea behind inheritance in Java is that you can create new **classes** that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the **IS-A relationship** which is also known as a parent-child relationship.

Why use inheritance in java

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Java Inheritance Example



As displayed in the above figure, **Programmer** is the subclass and **Employee** is the superclass. The relationship between the two classes is **Programmer IS-A Employee**. It means that **Programmer** is a type of **Employee**.

```

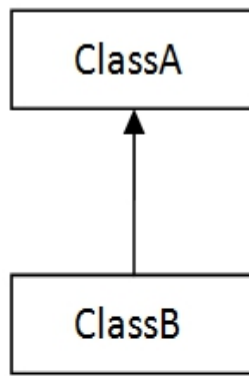
class Employee{
    float salary=80000;
    void m1(){
        System.out.println("Super class method");
    }
}
class Programmer extends Employee{
    int bonus=10000;
    void m2(){
        System.out.println("Sub class method");
    }
    public static void main(String args[]){
        Programmer p=new Programmer();
        System.out.println("Programmer salary is:"+p.salary); // 80000
        System.out.println("Bonus of Programmer is:"+p.bonus); // 10000
        p.m1();//
        p.m2();//
    }
}

```

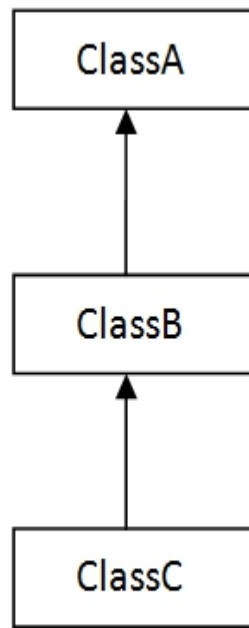
Types of inheritance in java

On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

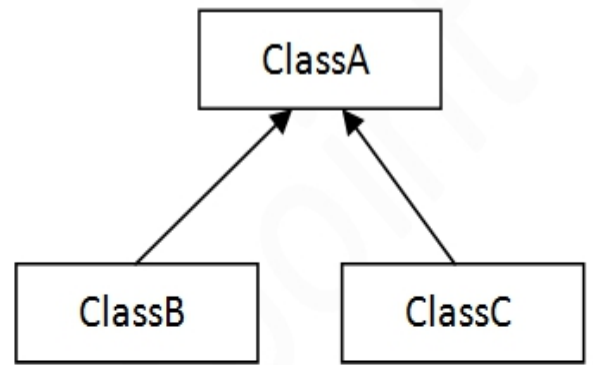
In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.



1) Single



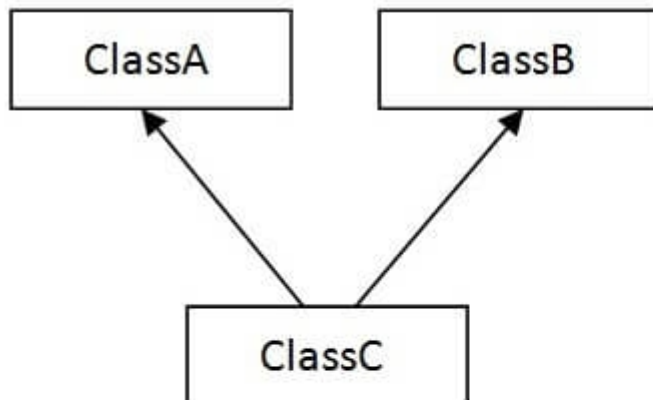
2) Multilevel



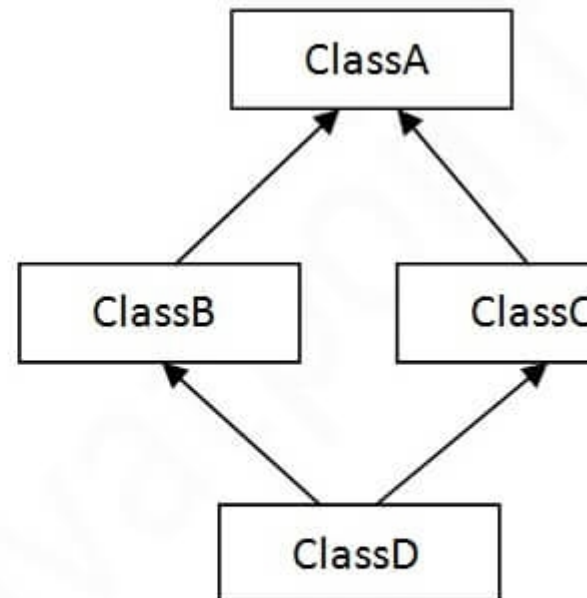
3) Hierarchical

Note: Multiple inheritance is not supported in Java through class.

When one class inherits multiple classes, it is known as multiple inheritance. For Example:



4) Multiple



5) Hybrid

When a class inherits another class, it is known as a single inheritance. In the example given below, Dog class inherits the Animal class, so there is the single inheritance.

File: TestInheritance.java

```
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();

}}
```

Execution of Constructor in Case of Inheritance

```
class A{

A(){

System.out.println("Super class constructor");

}

void show(){

System.out.println("Super class method");

}

}

class B extends class A{

B(){

System.out.println("Sub class constructor");
```

```
}  
  
}  
  
class C extends class B{  
  
    C(){  
  
        System.out.println("Grand son class constructor");  
  
    }  
  
}  
  
public static void main(String args[]){  
  
    C c = new C();  
  
    c.show();  
  
}  
  
}
```