



# Inheritance in Java

**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**.

## Terms used in Inheritance

- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

## The syntax of Java Inheritance

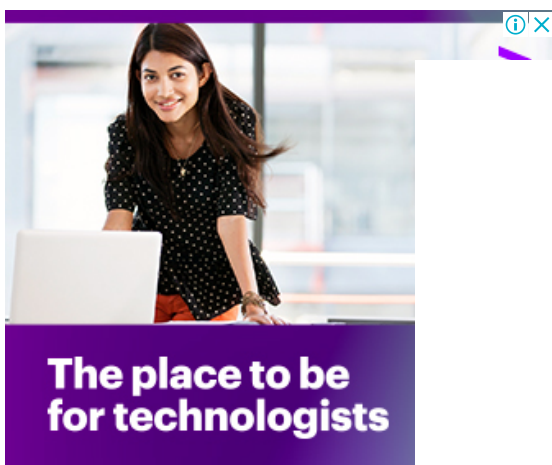
```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```

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.

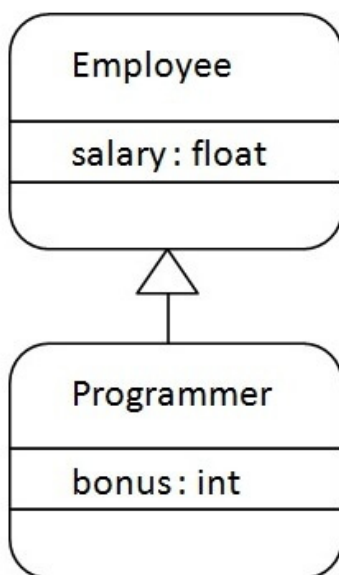


[ ]

In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.



## 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=40000;
}
class Programmer extends Employee{
int bonus=10000;
public static void main(String args[]){
Programmer p=new Programmer();
System.out.println("Programmer salary is:"+p.salary);
System.out.println("Bonus of Programmer is:"+p.bonus);
}
}
```

#### Test it Now

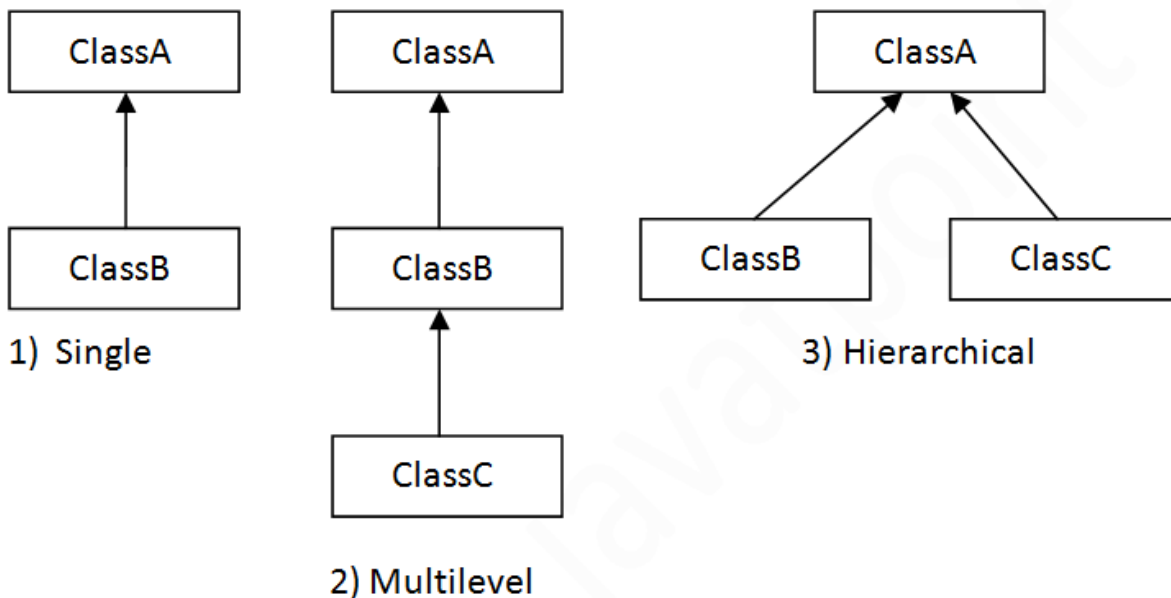
```
Programmer salary is:40000.0
Bonus of programmer is:10000
```

In the above example, Programmer object can access the field of own class as well as of Employee class i.e. code reusability.

## Types of inheritance in java

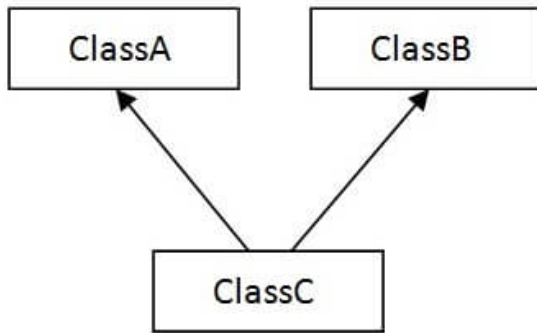
On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

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

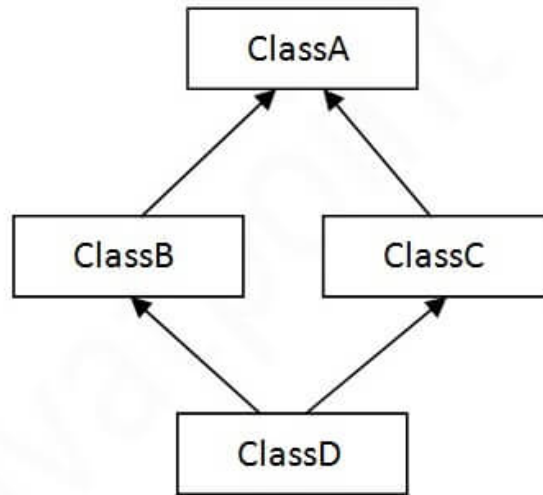


**Note:** Multiple inheritance is not supported in Java through class.

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



4) Multiple



5) Hybrid

## Single Inheritance Example

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();
}}
```

Output:

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

## Multilevel Inheritance Example

When there is a chain of inheritance, it is known as *multilevel inheritance*. As you can see in the example given below, BabyDog class inherits the Dog class which again inherits the Animal class, so there is a multilevel inheritance.

File: TestInheritance2.java

```
class Animal{  
    void eat(){System.out.println("eating...");}  
}  
class Dog extends Animal{  
    void bark(){System.out.println("barking...");}  
}  
class BabyDog extends Dog{  
    void weep(){System.out.println("weeping...");}  
}  
class TestInheritance2{  
    public static void main(String args[]){  
        BabyDog d=new BabyDog();  
        d.weep();  
        d.bark();  
        d.eat();  
    }  
}
```

Output:

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



## Hierarchical Inheritance Example

When two or more classes inherits a single class, it is known as *hierarchical inheritance*. In the example given below, Dog and Cat classes inherits the Animal class, so there is hierarchical inheritance.

File: TestInheritance3.java

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String args[]){
Cat c=new Cat();
c.meow();
c.eat();
//c.bark();//C.T.Error
}}
```

Output:

```
meowing...  
eating...
```

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

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

Since compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error.

```
class A{  
    void msg(){System.out.println("Hello");}  
}  
class B{  
    void msg(){System.out.println("Welcome");}  
}  
class C extends A,B{//suppose if it were  
  
    public static void main(String args[]){  
        C obj=new C();  
        obj.msg();//Now which msg() method would be invoked?  
    }  
}
```

Test it Now

Compile Time Error

← Prev

Next →



 [For Videos Join Our Youtube Channel: Join Now](#)


## Feedback


- Send your Feedback to [feedback@javatpoint.com](mailto:feedback@javatpoint.com)

## Help Others, Please Share




## Learn Latest Tutorials


 [Splunk tutorial](#)  
Splunk

 [SPSS tutorial](#)  
SPSS


 [Swagger tutorial](#)  
Swagger


 [T-SQL tutorial](#)  
Transact-SQL

 [Tumblr tutorial](#)  
Tumblr

 [React tutorial](#)  
ReactJS

 [Regex tutorial](#)  
Regex

 [Reinforcement learning tutorial](#)  
Reinforcement Learning

 [R Programming tutorial](#)  
R Programming

 [RxJS tutorial](#)  
RxJS



React Native  
tutorial

React Native



Python Design  
Patterns

Python Design  
Patterns



Python Pillow  
tutorial

Python Pillow



Python Turtle  
tutorial

Python Turtle



Keras tutorial

Keras

## Preparation



Aptitude

Aptitude



Logical  
Reasoning  
Reasoning



Verbal Ability  
Verbal Ability



Interview  
Questions  
Interview Questions



Company  
Interview  
Questions  
Company Questions

## Trending Technologies



Artificial  
Intelligence  
Tutorial

Artificial  
Intelligence



AWS Tutorial  
AWS



Selenium  
tutorial  
Selenium



Cloud  
Computing  
tutorial  
Cloud Computing



Hadoop tutorial  
Hadoop



ReactJS  
Tutorial  
ReactJS



Data Science  
Tutorial  
Data Science



Angular 7  
Tutorial  
Angular 7



Blockchain  
Tutorial  
Blockchain




Git Tutorial  
Git




Machine  
Learning Tutorial  
Machine Learning




DevOps  
Tutorial  
DevOps




DBMS tutorial  
DBMS




Data Structures  
tutorial  
Data Structures



DAA tutorial  
DAA




Operating  
System tutorial  
Operating System




Computer  
Network tutorial  
Computer Network



Compiler  
Design tutorial  
Compiler Design




Computer  
Organization and  
Architecture  
Computer  
Organization



Discrete  
Mathematics  
Tutorial  
Discrete  
Mathematics




Ethical Hacking  
Tutorial  
Ethical Hacking



Computer  
Graphics Tutorial  
Computer Graphics



Software  
Engineering  
Tutorial  
Software  
Engineering




html tutorial  
Web Technology




Cyber Security  
tutorial  
Cyber Security




Automata  
Tutorial  
Automata




C Language  
tutorial  
C Programming




C++ tutorial  
C++




Java tutorial  
Java




.Net  
Framework  
tutorial  
.Net



Python tutorial  
Python



List of  
Programs  
Programs



Control  
Systems tutorial  
Control System



Data Mining  
Tutorial  
Data Mining



Data  
Warehouse  
Tutorial  
Data Warehouse