**Types of inheritance in Java: Single,Multiple,Multilevel & Hybrid**

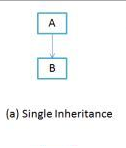
BY CHAITANYA SINGH | FILED UNDER: [OOPS CONCEPT](https://beginnersbook.com/category/oops-concept/)

<https://beginnersbook.com/2013/05/java-inheritance-types/>

Below are Various types of inheritance in Java. We will see each one of them one by one with the help of examples and flow diagrams.

**1) Single Inheritance**

**Single inheritance** is damn easy to understand. When a class extends another one class only then we  call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Here A is a **parent class** of B and B would be  a **child class** of A.

**[](https://beginnersbook.com/wp-content/uploads/2013/05/Single-Inheritance.png)**

**Single Inheritance example program in Java**

Class A

{

public void methodA()

{

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

}

}

Class B extends A

{

public void methodB()

{

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

}

public static void main(String args[])

{

B obj = new B();

obj.methodA(); //calling super class method

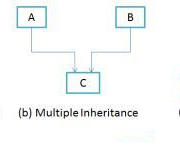
obj.methodB(); //calling local method

}

}

**2) Multiple Inheritance**

“**Multiple Inheritance**” refers to the concept of one class extending (Or inherits) more than one base class. The inheritance we learnt earlier had the concept of one base class or parent. The problem with “multiple inheritance” is that the derived class will have to manage the dependency on two base classes.

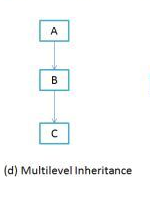
[](https://beginnersbook.com/wp-content/uploads/2013/05/Multiple-Inheritance.png)

Note 1: Multiple Inheritance is very rarely used in software projects. Using Multiple inheritance often leads to problems in the hierarchy. This results in unwanted complexity when further extending the class.

Note 2: Most of the new OO languages like **Small Talk, Java, C# do not support Multiple inheritance**. Multiple Inheritance is supported in C++.

**3) Multilevel Inheritance**

**Multilevel inheritance** refers to a mechanism in OO technology where one can inherit from a derived class, thereby making this derived class the base class for the new class. As you can see in below flow diagram C is subclass or child class of B and B is a child class of A. For more details and example refer – [Multilevel inheritance in Java](https://beginnersbook.com/2013/12/multilevel-inheritance-in-java-with-example/).

[](https://beginnersbook.com/wp-content/uploads/2013/05/Multilevel-Inheritance.png)

**Multilevel Inheritance example program in Java**

Class X

{

public void methodX()

{

System.out.println("Class X method");

}

}

Class Y extends X

{

public void methodY()

{

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

}

}

Class Z extends Y

{

public void methodZ()

{

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

}

public static void main(String args[])

{

Z obj = new Z();

obj.methodX(); //calling grand parent class method

obj.methodY(); //calling parent class method

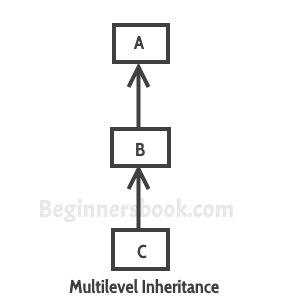
obj.methodZ(); //calling local method

}

}

# Multilevel inheritance in java with example

BY CHAITANYA SINGH | FILED UNDER: [OOPS CONCEPT](https://beginnersbook.com/category/oops-concept/)

When a class extends a class, which extends anther class then this is called **multilevel inheritance**. For example class C extends class B and class B extends class A then this [type of inheritance](https://beginnersbook.com/2013/05/java-inheritance-types/) is known as multilevel inheritance.  
Lets see this in a diagram:  


It’s pretty clear with the diagram that in Multilevel inheritance there is a concept of grand parent class. If we take the example of this diagram, then class C inherits class B and class B inherits class A which means B is a parent class of C and A is a parent class of B. So in this case class C is implicitly inheriting the properties and methods of class A along with class B that’s what is called multilevel inheritance.

To learn the basics of inheritance refer this tutorial: [Inheritance in Java](https://beginnersbook.com/2013/03/inheritance-in-java/)

## Multilevel Inheritance Example

In this example we have three classes –  Car, Maruti and Maruti800. We have done a setup – class Maruti extends Car and class Maruti800 extends Maruti. With the help of this Multilevel hierarchy setup our Maruti800 class is able to use the methods of both the classes (Car and Maruti).

class Car{

public Car()

{

System.out.println("Class Car");

}

public void vehicleType()

{

System.out.println("Vehicle Type: Car");

}

}

class Maruti extends Car{

public Maruti()

{

System.out.println("Class Maruti");

}

public void brand()

{

System.out.println("Brand: Maruti");

}

public void speed()

{

System.out.println("Max: 90Kmph");

}

}

public class Maruti800 extends Maruti{

public Maruti800()

{

System.out.println("Maruti Model: 800");

}

public void speed()

{

System.out.println("Max: 80Kmph");

}

public static void main(String args[])

{

Maruti800 obj=new Maruti800();

obj.vehicleType();

obj.brand();

obj.speed();

}

}

Output:

Class Car

Class Maruti

Maruti Model: 800

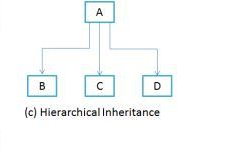
Vehicle Type: Car

Brand: Maruti

Max: 80Kmph

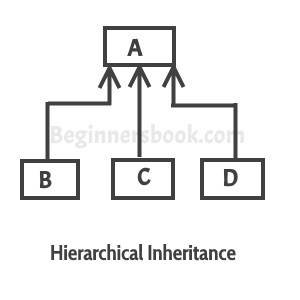
**4) Hierarchical Inheritance**

In such kind of inheritance one class is inherited by many**sub classes**. In below example class B,C and D **inherits** the same class A. A is **parent class (or base class)** of B,C & D. Read More at – [Hierarchical Inheritance in java with example program](https://beginnersbook.com/2013/10/hierarchical-inheritance-java-program/).

[](https://beginnersbook.com/wp-content/uploads/2013/05/Hierarchical-Inheritance.png)

# Hierarchical Inheritance in java with example program

BY CHAITANYA SINGH | FILED UNDER: [OOPS CONCEPT](https://beginnersbook.com/category/oops-concept/)

When more than one classes inherit a same class then this is called hierarchical inheritance. For example class B, C and D extends a same class A. Lets see the diagram representation of this:  


As you can see in the above diagram that when a class has more than one child classes (sub classes) or in other words more than one child classes have the same parent class then this [type of inheritance](https://beginnersbook.com/2013/05/java-inheritance-types/) is known as **hierarchical inheritance**.

If you find any difficulty in understanding the following example then refer this guide:  
[Java – Inheritance](https://beginnersbook.com/2013/03/inheritance-in-java/)

## Example of Hierarchical Inheritance

We are writing the program where class B, C and D extends class A.

class A

{

public void methodA()

{

System.out.println("method of Class A");

}

}

class B extends A

{

public void methodB()

{

System.out.println("method of Class B");

}

}

class C extends A

{

public void methodC()

{

System.out.println("method of Class C");

}

}

class D extends A

{

public void methodD()

{

System.out.println("method of Class D");

}

}

class JavaExample

{

public static void main(String args[])

{

B obj1 = new B();

C obj2 = new C();

D obj3 = new D();

//All classes can access the method of class A

obj1.methodA();

obj2.methodA();

obj3.methodA();

}

}

Output:

method of Class A

method of Class A

method of Class A

**5) Hybrid Inheritance**

In simple terms you can say that Hybrid inheritance is a combination of**Single** and **Multiple inheritance.** A typical flow diagram would look like below. A hybrid inheritance can be achieved in the java in a same way as multiple inheritance can be!! Using interfaces. yes you heard it right. By using **interfaces** you can have multiple as well as **hybrid inheritance** in Java.

# hybrid inheritance in java with example program

BY CHAITANYA SINGH | FILED UNDER: [OOPS CONCEPT](https://beginnersbook.com/category/oops-concept/)

A hybrid inheritance is a combination of more than one [types of inheritance](https://beginnersbook.com/2013/05/java-inheritance-types/). For example when class A and B extends class C & another class D extends class A then this is a hybrid inheritance, because it is a combination of single and hierarchical inheritance. Let me show you this diagrammatically:

C

↑

|

---------------

↑ ↑

| |

A B

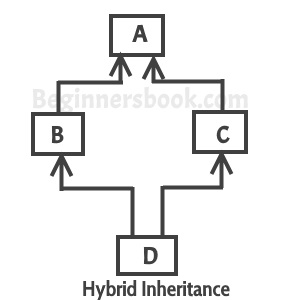
↑

|

D

If you want to learn the basics of inheritance, refer this guide: [java inheritance](https://beginnersbook.com/2013/03/inheritance-in-java/)

## Hybrid Inheritance in Java



It seems that because of this diagram people are finding it difficult to understand this topic because this diagram shows combination of hierarchical and multiple inheritance and [multiple inheritance](https://beginnersbook.com/2013/05/java-multiple-inheritance/) is not supported in java.  
The diagram is just for the representation, since multiple inheritance is not possible in java, It is not correct to show that as a part of hybrid inheritance. I will update the diagram whenever I get the time. You can refer the example that I have given at the beginning of post representing combination of single and hierarchical inheritance

## Hybrid Inheritance Example

Lets write this in a program to understand how this works:

C

↑

|

---------------

↑ ↑

| |

A B

↑

|

D

**Program:** This example is just to demonstrate the hybrid inheritance in Java. Although this example is meaningless, you would be able to see that how we have implemented two types of inheritance(single and hierarchical) together to form hybrid inheritance.  
Class A and B extends class C → [Hierarchical inheritance](https://beginnersbook.com/2013/10/hierarchical-inheritance-java-program/)  
Class D extends class A → Single inheritance

class C

{

public void disp()

{

System.out.println("C");

}

}

class A extends C

{

public void disp()

{

System.out.println("A");

}

}

class B extends C

{

public void disp()

{

System.out.println("B");

}

}

class D extends A

{

public void disp()

{

System.out.println("D");

}

public static void main(String args[]){

D obj = new D();

obj.disp();

}

}

Output:

D