Ans-1:

Inheritance Tutorial with Example in JAVA

Inheritance is a mechanism that allows the class to use the states and behavior of another class. In simple words a class derive field and methods from another class. The derived class in inheritance is called sub class (also called derived class or extended class, or child class) and the class from which it is derived is called super class (also called base class or a parent class).

Important Note: A super class can have any number of sub class in inheritance but sub class can only extend one super class. Multiple inheritance is not supported in JAVA.

Inheritance is one of the important concept of object oriented programming. The simple definition says inheritance provides mechanism that allows a class to inherit properties of another class.

Inheritance Concept: Inheritance means inherit the properties and behavior of existing object in a new object.

Way To Achieve Inheritance: It is achieved by deriving a new class from existing class using extends keyword. Example, `Class Child extends Base`

Explanation With Example:

- In Inheritance by default all data members and member functions of a parent class are available to child class if they are not private
- Inheritance defines **is-a** relationship between a super class(parent) and its sub class(child).
- **extends** keyword is used to show inheritance in java.

For example :

Suppose a class name Base.java

```
class Base
{
//Code of Base class
}
```

Another class Child.java use Inheritance to extends properties from Base class.

```
Class Child extends Base
```

```
{
//extends the properties of base class
}
```

In the above example, Child class will inherit field and methods of Base class.

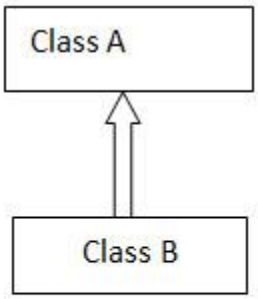
Types Of Inheritance

There are 3 types of inheritance:

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance

1. Single Inheritance:

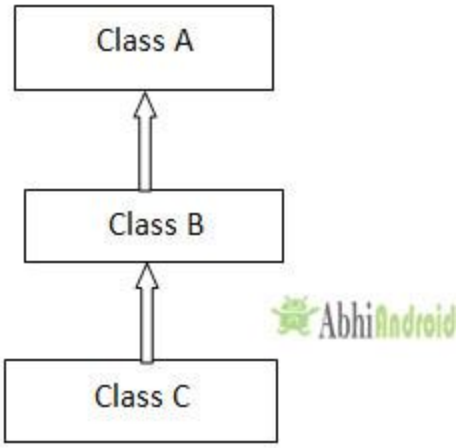
In Single inheritance one class is derived from one parent class. The below diagram shows single inheritance where Class B inherits from Class A.



Single Inheritance

2. Multilevel Inheritance:

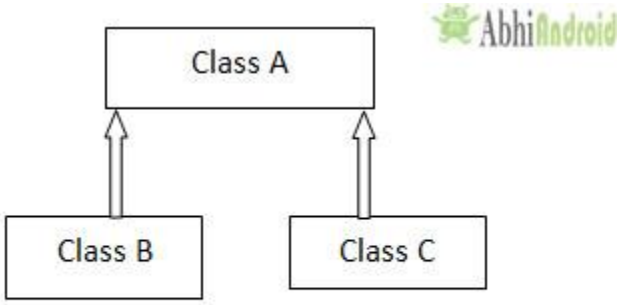
In multilevel inheritance there is a concept of grand parent class as shown in below diagram class C inherits class B and class B inherits class A.



Multilevel Inheritance

3. Hierarchical Inheritance:

In Hierarchical inheritance more than one sub classes is derived from a single parent class as shown in below diagram class B and C both are derived from single parent class A.



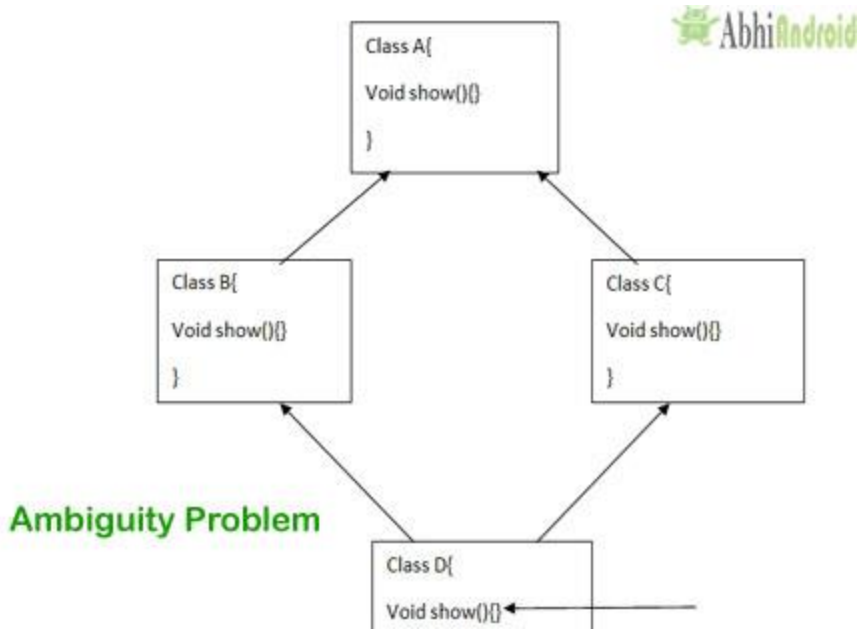
Heirarchical Inheritance

Important Note: Java does not support multiple Inheritance .

Why multiple inheritance is not supported in java:

To remove ambiguity:

Multiple inheritance is not supported in java as it causes ambiguity in few scenarios. The most common scenario is **Diamond problem.**



Consider the above diagram which shows multiple inheritance. In this class D extends both class B & C.Here class B & C inherit the same method of class A. Now the problem comes when class D is extending both class B & C and if class D wants to use same method then which method would be called? That's why multiple inheritance is not supported in java as to remove ambiguity.

Inheritance Example:

Below is the program to show you the use of inheritance in java. For coding this we have used eclipse IDE.

Example 1: Let's inherit some fields and methods in Child class from Base class.

Base class is having 2 fields and 1 method:

```
class Base
{
		int x=50;
		int y = 60;

//Addition method return integer value of x+y
		public int addition(){
				return x+y; //return 110
```

```
        }
}
```

Child class inherit addition method and x field from Base class:

```
public class Child extends Base
{

        int z;
        public void subtraction(){

                //addition method and x filed is inherited from Base Class
                z = addition() - x;
                System.out.println(z);
        }

}
```

In the Main class we create Child object and calls subtraction method on it.

```
public class InheritanceAbhiandroid {

public static void main(String[] args) {

Child child = new Child(); //Child object
child.subtraction();//Subtraction method called on child object

}

}
```

Output:

```
60
```

Example 2:

Base.java:

```
class Base
{
int x=50;
}
```

Child.java:

```
public class Child extends Base
{
 int x=20;
 void show()
 {
  System.out.println(x);
 }
}
```

InheritanceAbhiandroid.java

```
public class InheritanceAbhiandroid {

	public static void main(String[] args) {

		Child c = new Child();
		c.show();

	}

}
```

OUTPUT:

```
20
```

Important Note: The above program prints value of x=20 because priority always goes to local variables. So in show() method of child class,we can access member of parent class i.e base class using **super** keyword. It means if we want to print value of x=50 also from the same above program then by using super keyword.

Example 3 of Inheritance using super keyword:

Base.java:

```
class Base
{
int x=50;
}
```

Child.java:

```
class Child extends Base
{
int x=20;
void show()
{
System.out.println(x);
System.out.println(super.x);
}
}
public class InheritanceAbhiAndroid{


public static void main(String[] args)
{
Child c=new Child();
c.show();
}
}
```

OUTPUT:

```
20
```

Importance of Inheritance:

- **Reusuability of code**: It is one of the important feature of inheritance. It is a good way to reuse the already existing code rather than creating the same code again and again. This feature not only saves time and money as we are reusing the properties but it also increase reliability of code.
- **Method Overriding**: With the help of inheritance, it is possible to override the methods of base class so that base class method is easily used in derived class.

Inheritance Important Points to Remember:

- Whenever a parent class and a child class both are having same data members then this concept is known as **data hiding**.
- Whenever a parent class and a child class both are having ditto same functions then this concept is known as **method overriding.**
- Whenever a parent class and a child class both are having same static functions then this concept is known as **function hiding.**
- We cannot print **super**, there is a syntax error. Always data members of parent class is inherited by **super**
- If you make any non-static function of a class as **final** then it cannot be overridden by the child class that means to stop method overriding makes a function **final**.

Inheritance Quick Revision:

- Inheritance allows the class to use the states and behavior of another class using extends keyword
- Inheritance **is-a** relationship between a Base class and its child class.
- Multiple inheritance is not supported in JAVA.

Ans 2:

An abstract class is one that contains at least one abstract method. The class that inherits from this class has to implement the abstract methods of this parent class or itself be declared abstract.

Coming to its utility, it is used to define a common behavior for the classes that inherit from it. What to define is outlined in the abstract class whereas how to define or the implementation details are left to the child classes.

For example if you had an abstract class called shape, you could declare an abstract method (with only declaration no definition) to calculate the area.

From this class you can inherit classes like circle or square that would calculate the area of the corresponding shape.

Thus, implementing a common behavior among the classes.