What is Inheritance?

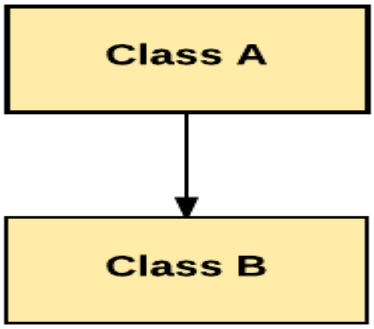
Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. Hence, inheritance facilitates Reusability and is an important concept of OOPs.

Types of Inheritance

There are Various types of inheritance in Java:

1. **Single Inheritance:**

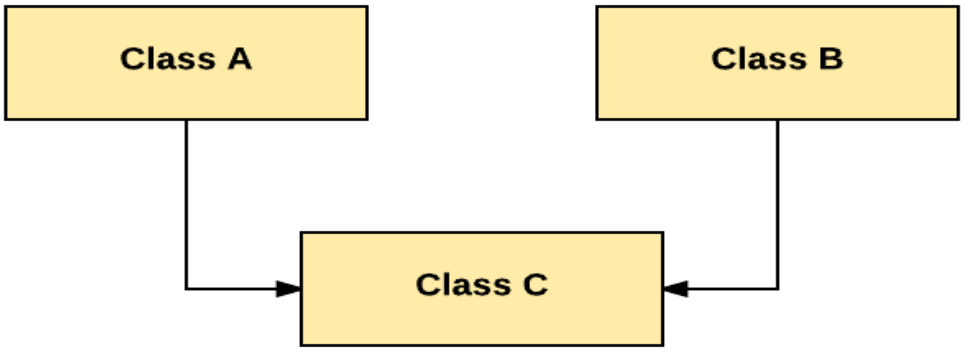
In Single Inheritance one class extends another class (one class only).



In above diagram, Class B extends only Class A. Class A is a super class and Class B is a Sub-class.

2. **Multiple Inheritance:**

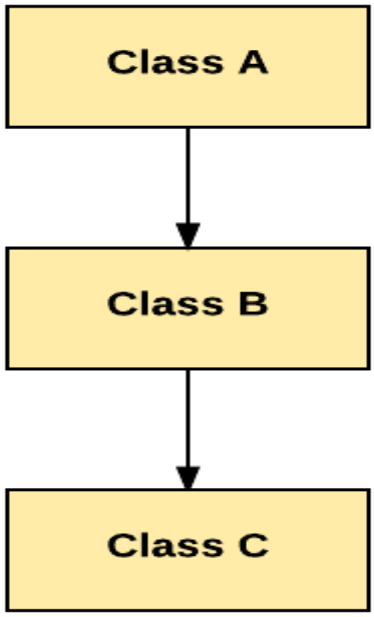
In Multiple Inheritance, one class extending more than one class. Java does not support multiple inheritance.



As per above diagram, Class C extends Class A and Class B both.

3. **Multilevel Inheritance:**

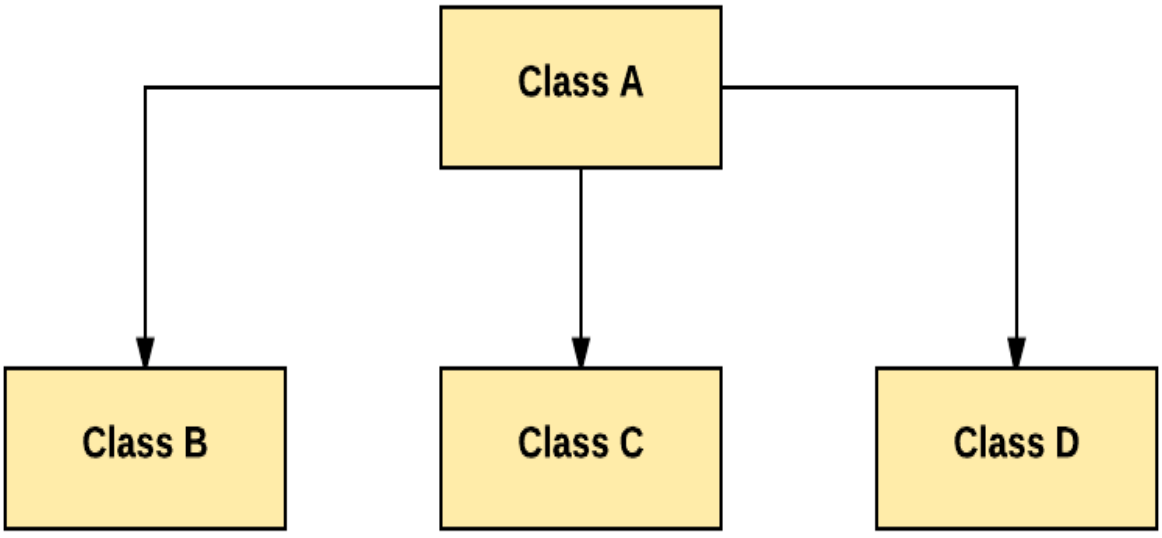
In Multilevel Inheritance, one class can inherit from a derived class. Hence, the derived class becomes the base class for the new class.



As per shown in diagram Class C is subclass of B and B is a of subclass Class A.

4. **Hierarchical Inheritance:**

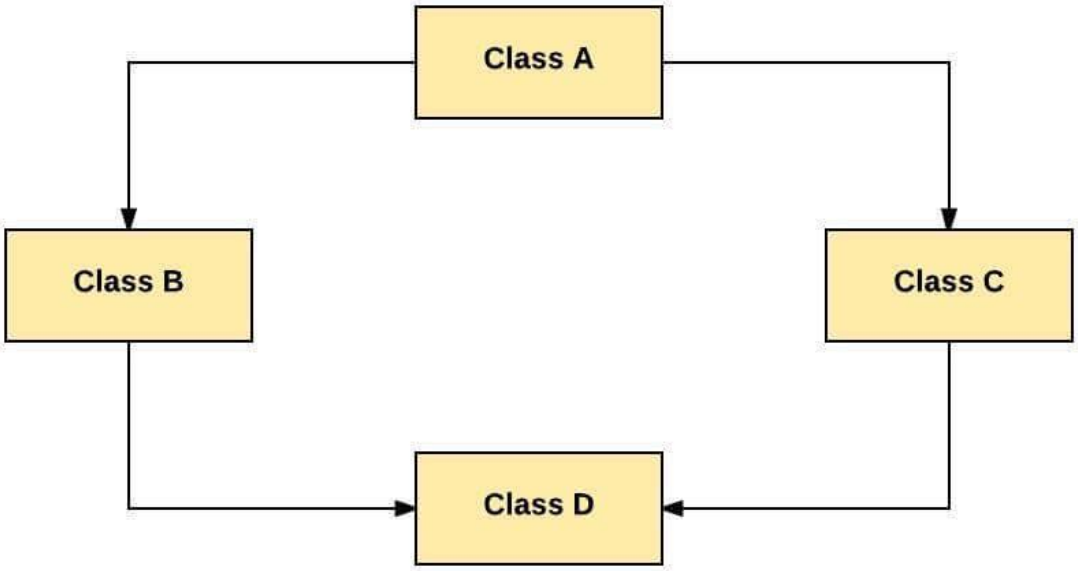
In Hierarchical Inheritance, one class is inherited by many sub classes.



As per above example, Class B, C, and D inherit the same class A.

5. **Hybrid Inheritance:**

Hybrid inheritance is a combination of Single and Multiple inheritance.



As per above example, all the public and protected members of Class A are inherited into Class D, first via Class B and secondly via Class C.

Inheritance in Java

In Java, when an "Is-A" relationship exists between two classes we use Inheritance

The parent class is termed super class and the inherited class is the sub class

The keyword "extend" is used by the sub class to inherit the features of super class

Inheritance is important since it leads to reusability of code

Java Inheritance Syntax:

```
class subClass extends superClass
{
  //methods and fields
}
```

```
import java.util.*;
import java.lang.*;
```

```
import java.io.*;
class one
{
    public void print_geek()
    {
        System.out.println("Geeks");
    }
}
class two extends one
{
    public void print_for()
    {
        System.out.println("for");
    }
}
class three extends one
{
    /*.....*/
}
// Drived class
public class Main
{
    public static void main(String[] args)
```

```
    {
        three g = new three();
        g.print_geek();
        two t = new two();
        t.print_for();
        g.print_geek();
    }
}
```

Output:

```
mca
for
mca
```