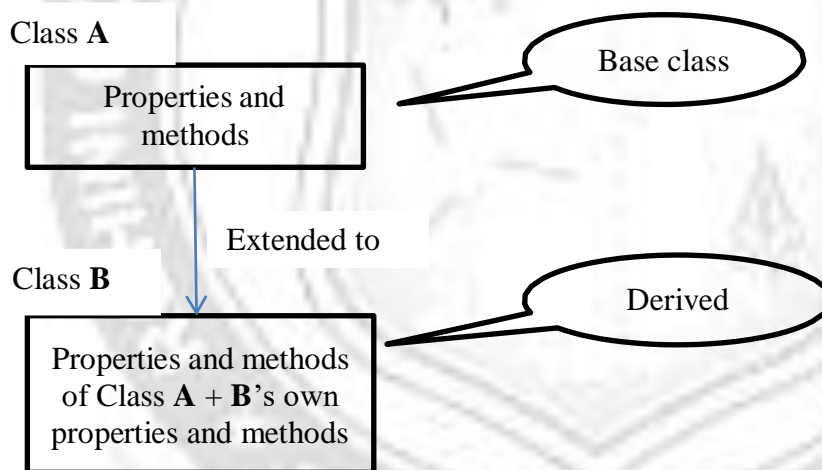


2.4 Inheritance

Definition:

Inheritance is a process of deriving a new class from existing class, also called as “extending a class”. When an existing class is extended, the new (inherited) class has all the properties and methods of the existing class and also possesses its own characteristics.

- ✓ The class whose property is being inherited by another class is called “**base class**” (or) “**parent class**” (or) “**super class**”.
- ✓ The class that inherits a particular property or a set of properties from the base class is called “**derived class**” (or) “**child class**” (or) “**sub class**”.



- ✓ Subclasses of a class can define their own unique behaviors and yet share some of the same functionality of the parent class.

➤ ADVANTAGES OF INHERITANCE:

- **Reusability of Code:**
 - ✓ Inheritance is mainly used for code reusability (Code reusability means that we can add extra features to an existing class without modifying it).
- **Effort and Time Saving:**
 - ✓ The advantage of reusability saves the programmer time and effort. Since the main code written can be reused in various situations as needed.
- **Increased Reliability:**
 - ✓ The program with inheritance becomes more understandable and easily maintainable as the sub classes are created from the existing reliably working classes.

➤ “extends” KEYWORD:

- ✓ Inheriting a class means creating a new class as an extension of another class.
- ✓ The **extends** keyword is used to inherit a class from existing class.
- ✓ The general form of a **class** declaration that inherits a superclass is shown here:

✓ **Syntax:**

```
[access_specifier] class subclass_name extends superclass_name
{
    // body of class
}
```

Characteristics of Class Inheritance:

1. A class cannot be inherited from more than one base class. Java does not support the inheritance of multiple super classes into a single subclass.
2. Sub class can access only the non-private members of the super class.
3. Private data members of a super class are local only to that class. Therefore, they can't be accessed outside the super class, even sub classes can't access the private members.
4. Protected features in Java are visible to all subclasses as well as all other classes in the same package.

✓ **Example:**

```
class Vehicle
{
    String brand;
    String color;
}
class Car extends Vehicle
{
    int totalDoor;
}

class Bike extends Vehicle
{
}
```

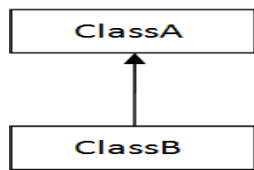
In the above example, Vehicle is the **super class** or base class that holds the common property of Car and Bike. Car and Bike is the **sub class** or derived class that inherits the property of class Vehicle **extends** is the keyword used to inherit a class.

TYPES OF INHERITACE:

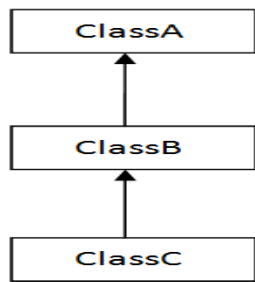
1. Single Inheritance
2. Multilevel Inheritance
3. Multiple Inheritance

Note: The following inheritance types are not directly supported in Java.

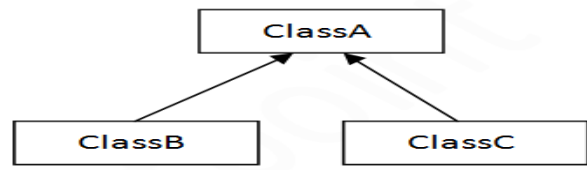
4. Hierarchical Inheritance
5. Hybrid Inheritance



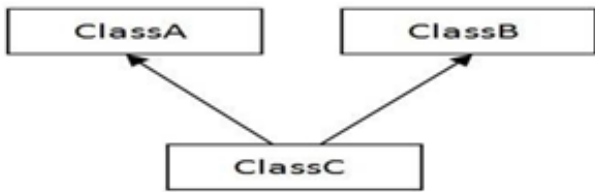
1) Single



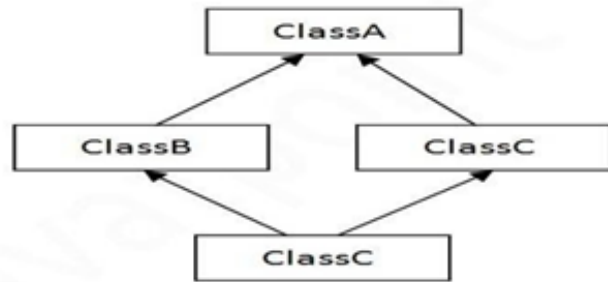
2) Multilevel



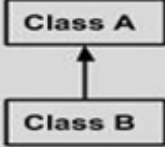
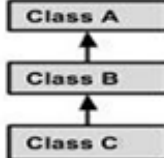
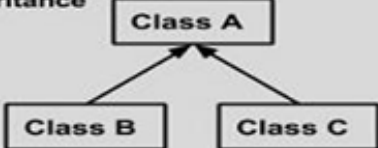
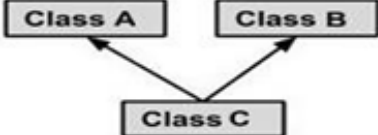
3) Hierarchical



4) Multiple



5) Hybrid

<p>Single Inheritance</p> 	<pre> public class A { } public class B extends A { } </pre>
<p>Multi Level Inheritance</p> 	<pre> public class A {} public class B extends A {.....} public class C extends B {.....} </pre>
<p>Hierarchical Inheritance</p> 	<pre> public class A {} public class B extends A {.....} public class C extends A {.....} </pre>
<p>Multiple Inheritance</p> 	<pre> public class A {} public class B {.....} public class C extends A,B { } // Java does not support multiple Inheritance </pre>