

# The Terminologies#

As we know that a new class is created *based* on an *existing* class in Inheritance, hence we use the terminology below for the new class and the existing class:

- **SuperClass (Mother Class or Base Class):** This class allows the *re-use* of its `non-private` members in another class.
- **SubClass (Child Class or Derived Class):** This class is the one that *inherits* from the superclass.

💡 A *child* class has **all non-private** characteristics of the *mother* class.

## What does a Child have?#

An object of the child class can use:

- All `non-private` members defined in the **child** class.
- All `non-private` members defined in the **mother** class.

⚠️ Some classes cannot be inherited. Such classes are defined with the keyword, `final`. An example of such a class is the built-in `Integer class` - this class cannot have derived classes.

## The `extends` Keyword#

In Java, we have to use the keyword `extends` to implement inheritance:

```
SubClass extends SuperClass{
//contents of SubClass
}
```

Let’s take an example of a `Vehicle class` as a *base class* and implement a `Car class` that will extend from this `Vehicle class`. As a *Car IS A, Vehicle* the implementation of inheritance relation between these classes will stand valid.

```
1 // Base Class Vehicle
2 class Vehicle {
3
4     // Private Fields
5     private String make;
6     private String color;
7     private int year;
8     private String model;
9
10
11     // Parameterized Constructor
12     public Vehicle(String make, String color, int year, String model) {
13         this.make = make;
14         this.color = color;
15         this.year = year;
16         this.model = model;
17     }
18
19     // public method to print details
20     public void printDetails() {
21         System.out.println("Manufacturer: " + make);
22         System.out.println("Color: " + color);
23         System.out.println("Year: " + year);
24         System.out.println("Model: " + model);
25     }
26
27 }
28
```

Run Save Reset

In the code above, ignore the **line 37** for now, you will get to know about it in the next lesson.

**Note:** In Java, a class can extend from only one other class at a time and a class cannot extend itself.