INHERITANCE IN JAVA

- Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.
- The idea behind inheritance in Java is that you can create new classes that are built upon existing classes.
- When you inherit from an existing class, you can reuse methods and attributes of the parent class. Moreover, you can add new methods and attributes in your current class also.
- Inheritance is also known as a parent-child relationship.

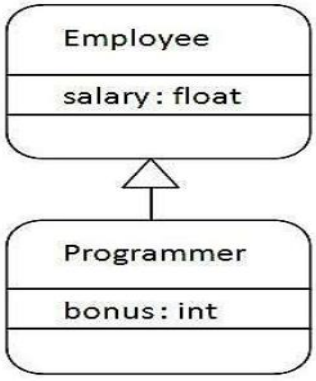
Terms used in Inheritance

- **Class**: A class is a template or blueprint from which objects are created.
- **Sub Class/Child Class**: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class**: Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability**: As the name specifies, reusability is a mechanism which facilitates you to reuse the attributes and methods of the existing class when you create a new class. We can use the same attributes and methods already defined in the previous class.

The syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

- The extends keyword indicates that you are making a new class that derives from an existing class.
- The meaning of "extends" is to increase the functionality.
- In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.



Types of inheritance in java

- On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.