BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on Inheritance in Java.

Inheritance in Java:-

Inheritance is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of Object Oriented programming system.

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

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 fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the **IS-A relationship** which is also known as a **parent-child relationship**.

Super Class/Parent Class: Superclass is the class from where a subclass inherits the features. The class whose properties are inherited is known as **superclass.** It is also called a base class or a parent class.

Sub Class/Child Class: Subclass is a class which inherits the other class. The class which inherits the properties of other is known as **subclass.** It is also called a derived class,

extended class, or child class.

Why use inheritance in java:-

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

extends Keyword:-

extends is the keyword used to inherit the properties of a class.

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.

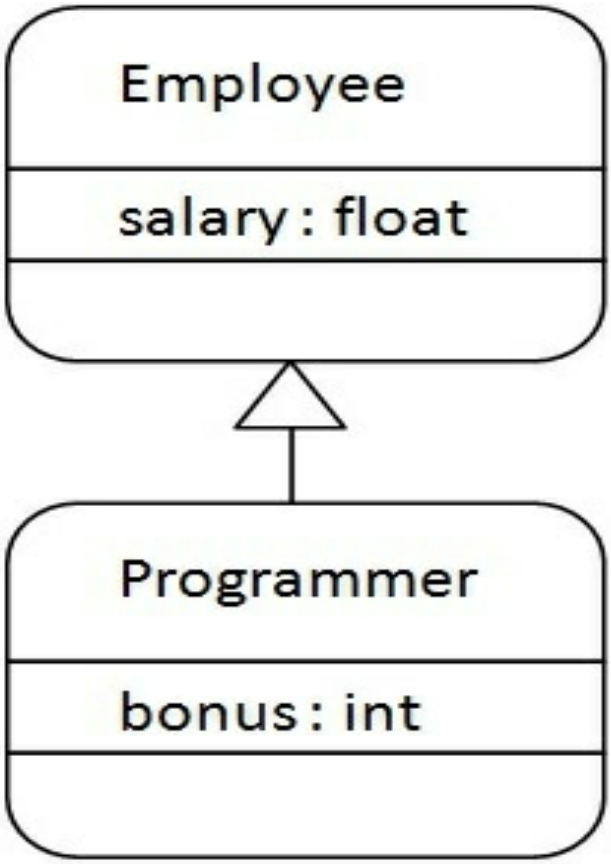
Following is the syntax of extends keyword.

```
class Super
{
   .....
   .....
}
class Sub extends Super
{
   .....
   .....
}
```

Syntax of Java Inheritance:-

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

Inheritance Example:-



As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationship between the two classes is **Programmer IS-A Employee**. It means that Programmer is a type of Employee.

```
class Employee
{
 float salary=40000;
}
class Programmer extends Employee
{
 int bonus=10000;
 public static void main(String args[])
{
   Programmer p=new Programmer();
   System.out.println("Programmer salary is:"+p.salary);
   System.out.println("Bonus of Programmer is:"+p.bonus);
}
}
```

Output:-

Programmer salary is:40000.0

Bonus of programmer is:10000

In the above example, Programmer object can access the field of own class as well as of Employee class i.e. code reusability.