Java Inheritance

By Umesh Sir

Contact us 7758094241

JavaScript

- ★ What is mean by Inheritance?
- ★ Important Terminology?
- ★ Syntax?
- ★ Simple Example?

★ What is mean by Inheritance?

- ★ The Inheritance is a process of obtaining the data members and methods from one class to another class, plus can have its own is known as inheritance. It is one of the fundamental features of object-oriented programming.
- ★ IS-A relationship between a superclass and its subclasses.

★ Important Terminology?

- ★ Super Class: The class whose features are inherited is known as a superclass (or a base class or a parent class).
- ★ Sub Class: The class that inherits the other class is known as a subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.

★ Syntax?

```
class derived(Sub or Child)-class extends base (Super or Parent)-class
{
 //methods and fields
}
Class A{ }
Class B extends A{ }
```

★ Simple Example?

- Common Employee (Parent) Class

Employee.java

```
1  package infosys;
2
3  public class Employee {
4
5       private  int empId;
6       private String Name;
7       private float salary;
8       private String mobNo;
9
10      public int getEmpId() {
11          System.out.println("Parent Class id");
12          return empId;
13      }
14      public void setEmpId(int empId) { this.empId = empId; }
15      public String getName() {
16          return Name;
17      }
18      public void setName(String name) { Name = name;}
19
20      public float getSalary() {
21          System.out.println("Parent Class Salary");
22          return salary;
23      }
24      public void setSalary(float salary) { this.salary = salary; }
25      public String getMobNo() { return mobNo; }
26      public void setMobNo(String mobNo) { this.mobNo = mobNo; }
27 }
28
```

SWDeveloper Employee Class.

Bonus Is additional for

sw developer.

SWDeveloper.java

```
1  package infosys.developer;
2
3  import infosys.Employee;
4
5  public class SWDeveloper extends Employee{
6
7      private float bonus;
8
9      public float getSWSalary() {
10
11         return getSalary()+ getBonus();
12     }
13
14     public float getBonus() {
15         return bonus;
16     }
17
18     public void setBonus(float bonus) {
19         this.bonus = bonus;
20     }
21
22 }
23
```

Tester Employee Class.

shiftAllowance is additional

For Tester.

Tester.java

```java
package infosys.tester;

import infosys.Employee;

public class Tester extends Employee{

    float shiftAllownce;


    public float getTesterSalary() {

        return getSalary() + getShiftAllownce();
    }

    public float getShiftAllownce() {
        return shiftAllownce;
    }

    public void setShiftAllownce(float shiftAllownce) {
        this.shiftAllownce = shiftAllownce;
    }



}
```

Test Class

EmpTest.java

```
1  package infosys;
2
3  import infosys.developer.SWDeveloper;
4  import infosys.tester.Tester;
5
6  public class EmpTest {
7
8      public static void main(String[] args) {
9
10         SWDeveloper dev = new SWDeveloper();
11         dev.setEmpId(10);
12         dev.setMobNo("45787956454");
13         dev.setName("Umesh");
14         dev.setSalary(10000);
15         dev.setBonus(2000);
16
17         System.out.println("SW Developer Id is " + dev.getEmpId());
18         System.out.println("SW Developer Salary is " +dev.getSWSalary());
19
20         Tester test = new Tester();
21         test.setEmpId(50);
22         test.setMobNo("564879855");
23         test.setName("Suyog");
24         test.setSalary(10000);
25         test.setShiftAllownce(5000);
26
27         System.out.println("SW Tester Id is " + test.getEmpId());
28         System.out.println("SW Tester Salary is " +test.getTesterSalary());
29
30     }
31 }
32
```

Thank you