Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**8**
Student Name:	**Bhagyesh Subhash Manjrekar**
Roll No :	**07**

Title:

1. To perform Multilevel Inheritance in JAVA. Create a Person class representing name, age and address. Inherit person class to employee class with emp ID and salary factor. Inherit the Employee class to programmer class with technical skills and hike attributes. Implement valid methods to input the details from the user in the main method and display for 3 programmers.
2. To perform Hierarchical Inheritance in JAVA. Create an Employee class with attributes EmpID and EmpSalary. Also create necessary methods/constructors to accept these values from the user. Create classes permenantEmployee and TemporaryEmployee which will be derived classes of Employee. Mention hike attribute in these derived classes and calculate the total salary using generate_salary() method for respective types of employees. Objects of the derived classes should be created and salaries for the permanent and temporary employees should be calculated and displayed on the screen.

Learning Objective:

- Students will be able to perform multilevel inheritance using JAVA.
- Students will be able to perform hierarchical inheritance using JAVA

Learning Outcome:

- To understand how to use the private members using friend function and friend class.

Course Outcome:

- Comprehend building blocks of OOPs language, inheritance, package and interfaces.

Theory:

1. Explain in details about various inheritance types supported in JAVA:

Ans:

- **Definition:**

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours 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 fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance is used for **method overriding** and for code **reusability**.

- **Terms related to inheritance in JAVA:**

Sub Class: It is a class which inherits the other class. It is also called a derived class, extended class, or child class.

Super Class: It is a class from where a subclass inherits the features. It is also called a base class or a parent class.

Reusability: Reusability is a mechanism which facilitates us to reuse the fields and methods of the existing class when we create a new class. We can use the same fields and methods already defined in the previous class.

Syntax:

```
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.

- **Types of Inheritance in JAVA:**

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

In java programming, multiple and hybrid inheritance is supported through interface only.

(1) Single Inheritance:

When a one class inherits another class, it is known as a single inheritance.

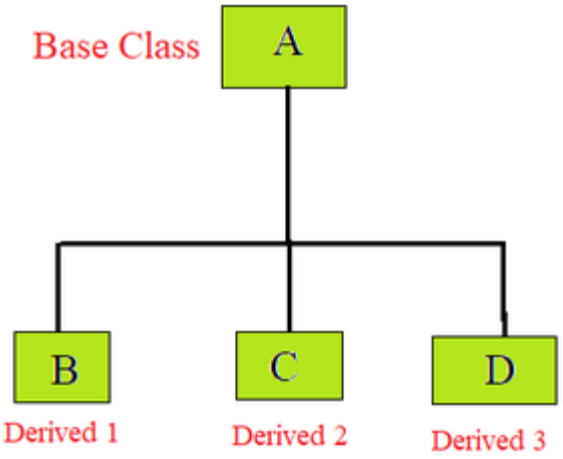
(2) Multilevel Inheritance:

In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class. In Java, a class cannot directly access the grandparent's members.

When there is a chain of inheritance, it is known as multilevel inheritance.

(3) Hierarchical Inheritance:

When two or more classes inherits a single class, it is known as hierarchical inheritance.



To reduce the complexity and simplify the language, multiple inheritance is not supported in java. But it can be supports if we use it through interfaces, In Multiple inheritances, one class can have more than one superclass and inherit features from all parent classes.

Program 1: To perform Hierarchical Inheritance in JAVA. Create an Employee class with attributes EmpID and EmpSalary. Also create necessary methods/constructors to accept these values from the user. Create classes permenantEmployee and TemporaryEmployee which will be derived classes of Employee. Mention hike attribute in these derived classes and calculate the total salary using generate_salary() method for respective types of employees. Objects of the derived classes should be created and salaries for the permanent and temporary employees should be calculated and displayed on the screen.

Algorithm :

Step 1: Start

Step 2: Create class employee with attributes EmpID and EmpSalary, using method of Hierarchical Inheritance.

Step 3: Create a constructor for employee class to accept values from user.

Step 4: Create classes permenantEmployee and TemporaryEmployee which will be derived classes of Employee.

Step 5: Create hike attribute in these derived classes and calculate the total salary using generate_salary() method for respective types of employees.

Step 6: Crate object of the derived classes and print the output for salaries of both types of employees.

Step 7: Stop

Program:

```
package inheritence_java;


class employee

{

float salary = 25000;

void display()
```

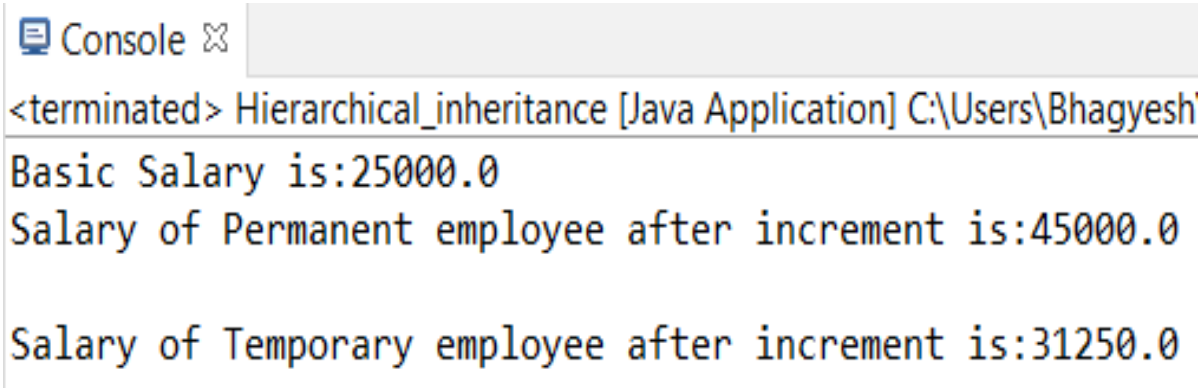
```
{
System.out.println("Basic Salary is:" + salary);
}
}
class Permanentemp extends employee
{
double hike = 0.8;
void incrementsalary()
{
super.display();
System.out.println("Salary of Permanent employee after increment is:" +
(salary+(salary *hike)) );
}
}
class Temporaryemp extends employee
{
double hike = 0.25;
void incrementsalary()
{
System.out.println("");
System.out.println("Salary of Temporary employee after increment is:" +
(salary+(salary *hike)) );
}
}
public class Hierarchical_inheritance
{
```

```
public static void main(String args[])
{
Permanentemp per = new Permanentemp();
Temporaryemp temp = new Temporaryemp();
per.incrementsalary();
temp.incrementsalary();
}
}
```

Input given:

Basic Salary is:25000.0

Output Screenshot:



Program 2: To perform Multilevel Inheritance in JAVA. Create a Person class representing name, age and address. Inherit person class to employee class with emp ID and salary factor. Inherit the Employee class to programmer class with technical skills and hike attributes. Implement valid methods to input the details from the user in the main method and display for programmers.

Algorithm:

Step 1: Start

Step 2: Create a Person class representing name, age and address, take input from user.

Step 3: Inherit person class to employee class, have parameters as employee id, salary factor etc.

Step 4: Inherit the Employee class to programmer class, have parameters as profession, increment, hike etc.

Step 5: Create object of employee, and programmer class and print the data of the programmer/employee.

Step 6: Stop

Program:

```
package inheritence_java;


import java.util.Scanner;

class person

{

String name, address;

int age;

Scanner in = new Scanner(System.in);
```

```
void display()
{
System.out.println("***Details of the Programmer***");
System.out.println("Enter your name:");
name=in.next();
System.out.println("Enter your age:");
age=in.nextInt();
System.out.println("Enter your residental address:");
address=in.next();
}
}
class employee1 extends person
{
int empid;
float salary=40000 ;
Scanner in= new Scanner(System.in);
void getdata()
{
System.out.println("Employee ID is:");
empid=in.nextInt();
}
void getsalary()
{
System.out.println("Base Salary is:" +salary);
}
}
```

```
class programmer extends employee1
{
Scanner in = new Scanner(System.in);
String Profession;
double hike=0.5 ;
void printdata()
{
System.out.println("Technical Skills/Profession: ");
Profession=in.next();
}
void hikesalary()
{
super.getsalary();
System.out.println("Salary after increment is: " +(salary+(salary*hike)) );
}
}
public class Multilevel_inheitence
{
public static void main(String[] args)
{
// TODO Auto-generated method stub
employee1 emp = new employee1();
programmer p = new programmer();
emp.display();
emp.getdata();
p.printdata();
```

```
p.hikesalary();
}
}
```

Input given:

Base Salary is: 40000

hike=0.5

Output Screenshot: