

**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**

<b>Lab Number:</b>	<b>8</b>
<b>Student Name:</b>	<b>Shruti Gokhale</b>
<b>Roll No :</b>	<b>40</b>

**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 permanentEmployee 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:**

<b>ECL304.2</b>	Comprehend building blocks of OOPs language, inheritance, package and interfaces.
-----------------	---

**Theory:**

- Explain in details about various inheritance types supported in JAVA
1. single inheritance: A sub-class is derived from only one super class. It inherits the properties and behavior of a single-parent class.
  2. multi-level inheritance: A class is derived from a class which is also derived from another class is called multi-level inheritance. In simple words, we can say that a class that has more

**Faculty: Ms. Deepali Kayande**

**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**

than one parent class is called multi-level inheritance. The classes must be at different levels. Hence, there exists a single base class and single derived class but multiple intermediate base classes.

3. hierarchical inheritance: If a number of classes are derived from a single base class, it is called hierarchical inheritance.
4. hybrid inheritance: It consist of more than one. Hybrid inheritance is the combination of two or more types of inheritance.
5. Java does not support multiple inheritances due to ambiguity.

<b>Algorithm 1:</b>	<ol style="list-style-type: none"> <li>1) Start</li> <li>2) Create class Person</li> <li>3) Define attributes and method display() and getDetails()</li> <li>4) Create child class Employee</li> <li>5) Define attriutes salary EmpID and methods display() &amp; getDetails()</li> <li>6) Create another child class Programmer</li> <li>7) Define attributes hike, skills and methods display() &amp; getDetails()</li> <li>8) In main class, create 3 objects for 3 programmers</li> <li>9) Display output</li> <li>10) Stop</li> </ol>
<b>Program:</b>	<pre> package enheritance; import java.util.Scanner; class Person{ Scanner in = new Scanner(System.in); String name; String address; int age; Person() { name = ""; address = ""; age = 0; } void display() { System.out.println("Name : "+name); System.out.println("Age : "+age);  System.out.println("Address : "+address); } void getDetails() { System.out.println("Enter name : "); name = in.nextLine(); System.out.println("Enter address : "); address = in.nextLine(); System.out.println("Enter age : "); age = in.nextInt(); } </pre>

**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**

```
}  
class Employee extends Person{  
    int emplID; double salary;  
    Employee()  
    {  
        emplID = 0;  
        salary = 0.0;  
    }  
    void getDetails()  
    {  
        super.getDetails();  
        System.out.println("Enter Employee ID : ");  
        emplID = in.nextInt();  
        System.out.println("Enter base salary : ");  
        salary = in.nextDouble();  
    }  
    void display()  
  
    {  
        super.display();  
        System.out.println("Employee ID : "+emplID);  
        System.out.println("Base Salary : Rs."+salary);  
    }  
}  
public class Programmer extends Employee  
{  
    double hike; String skills;  
    Programmer()  
    {  
        hike = 0;  
        skills = "";  
    }  
    void getDetails()  
    {  
        super.getDetails();  
        System.out.println("Enter salary hike : ");  
        hike = in.nextDouble();  
        System.out.println("Enter technical skills : ");  
        in.nextLine();  
        skills = in.nextLine();  
    }  
    void display()  
    {  
        super.display();  
  
        System.out.println("Salary Hike : Rs."+hike);  
        System.out.println("Total salary : Rs."+(salary+hike));  
        System.out.println("Technical skills : "+skills);  
    }  
}
```

**Faculty: Ms. Deepali Kayande**

**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**

	<pre> public static void main(String args[]) {     System.out.println("Enter details for 1st programmer");     Programmer obj1 = new Programmer();     obj1.getDetails();     System.out.println("Enter details for 2nd programmer");     Programmer obj2 = new Programmer();     obj2.getDetails();     ;     System.out.println("Enter details for 3rd programmer");     Programmer obj3 = new Programmer();     obj3.getDetails();     System.out.println("\nDetails of 1st programmer");     obj1.display();     System.out.println("\nDetails of 2nd programmer");     obj2.display();     System.out.println("\nDetails of 3rd programmer");     obj3.display(); } } </pre>
<b>Input given:</b>	<p>Enter details for 1st programmer</p> <p>Enter name : Lili</p> <p>Enter address : lala</p> <p>Enter age : 24</p> <p>Enter Employee ID : 10</p> <p>Enter base salary : 1000</p> <p>Enter salary hike : 12</p> <p>Enter technical skills : none</p> <p>Enter details for 2nd programmer</p> <p>Enter name : Jug</p> <p>Enter address : lalaa</p> <p>Enter age : 23</p> <p>Enter Employee ID : 11</p> <p>Enter base salary : 1000</p> <p>Enter salary hike : 12</p> <p>Enter technical skills : none</p> <p>Enter details for 3rd programmer</p>

**Faculty: Ms. Deepali Kayande**

**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**

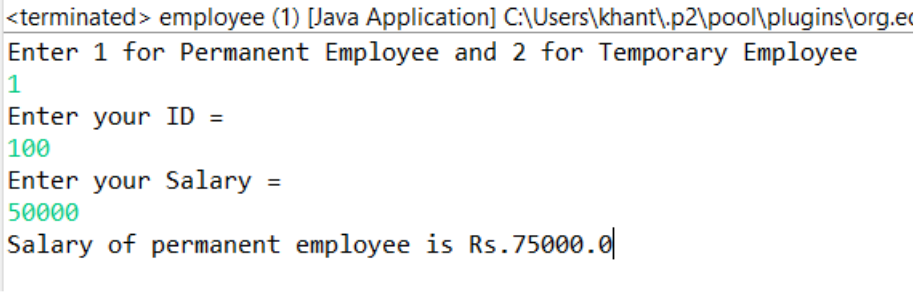
	<p>Enter name : Ronnie Enter address : lalla Enter age : 20 Enter Employee ID : 12 Enter base salary : 1000 Enter salary hike : 12 Enter technical skills : none</p>
<b>Output Screenshot:</b>	<pre>Name : Lili Age : 24 Address : lala Employee ID: 10 Base Salary : Rs.1000.0 Salary Hike : Rs.12.0 Total salary : Rs.1012.0 Technical skills : none  Details of 2nd programmer Name : Jug Age : 23 Address : lalaa Employee ID: 11 Base Salary : Rs.1000.0 Salary Hike : Rs.12.0 Total salary : Rs.1012.0 Technical skills : none  Details of 3rd programmer Name : Ronnie Age : 20 Address : lalla Employee ID: 12 Base Salary : Rs.1000.0 Salary Hike : Rs.12.0 Total salary : Rs.1012.0 Technical skills : none</pre>

**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**

<b>Algorithm 2:</b>	<ol style="list-style-type: none"><li>1. Start</li><li>2. create class employee1, define attributes and methods setdetails()</li><li>3. create child classes PermanentEmp and TemporaryEmp</li><li>4. define attributes and method generatesalary() in both the classes</li><li>5. Create main function</li><li>6. Give the user 2 choices of permanent or temporary employee</li><li>7. create object in main function according to the case selected</li><li>8. print the output</li><li>9. Stop</li></ol>
<b>Program :</b>	<pre>import java.util.Scanner; class employee1 { int EmpID; float salary; void setdetails() { Scanner t = new Scanner(System.in); System.out.println("Enter your ID ="); EmpID= t.nextInt(); System.out.println("Enter your Salary ="); salary= t.nextFloat(); } } class PermanentEmp extends employee1{ double hike = 0.5; void generatesalary() { System.out.println("Salary of permanent employee is Rs." +(salary + (salary * hike)) ); } } class TemporaryEmp extends employee1{ double hike = 0.2; void generatesalary() { System.out.println("Salary of temporary employee is Rs." +(salary + (salary * hike)) ); } } public class employee { public static void main(String args[]) { Scanner in = new Scanner(System.in); System.out.println("Enter 1 for Permanent Employee and 2 for Temporary Employee");</pre>

**Faculty: Ms. Deepali Kayande**

**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**

	<pre> int choice = in.nextInt(); switch(choice) { case 1 : PermanentEmp p = new PermanentEmp(); p.setdetails(); p.generatesalary(); break; case 2: TemperoryEmp t = new TemperoryEmp(); t.setdetails(); t.generatesalary(); break; default: System.out.println("Invalid choice"); } } } </pre>
<b>Input Given:</b>	Enter 1 for Permanent Employee and 2 for Temporary Employee 1 Enter your ID = 100 Enter your Salary = 50000
<b>Output Screenshot:</b>	 <pre> &lt;terminated&gt; employee (1) [Java Application] C:\Users\khant\p2\pool\plugins\org.e Enter 1 for Permanent Employee and 2 for Temporary Employee 1 Enter your ID = 100 Enter your Salary = 50000 Salary of permanent employee is Rs.75000.0 </pre>