|  |  |
| --- | --- |
| **Lab Number:** | **8** |
| **Student Name:** | **Omkar Santosh Mundhe** |
| **Roll No :** | **20** |

# 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:**

|  |  |
| --- | --- |
| **ECL304.2** | Comprehend building blocks of OOPs language, inheritance, package and interfaces. |

**Theory:**

* + Explain in details about various inheritance types supported in JAVA.

[Inheritance](https://www.javatpoint.com/inheritance-in-java) is a mechanism of driving a new class from an existing class. The existing (old) class is known a**s** base class **or** super class **or** parent class**.** The new class is known as

a derived class or sub class or child 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.

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 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.

|  |  |
| --- | --- |
| **Algorithm :** | STEP 1: Start  STEP 2:Create class Person  STEP 3:Define attributes and method display() and getDetails() STEP 4:Create child class Employee  STEP 5:Define attriutes salary EmpID and methods display() & getDetails() STEP 6:Create another child class Programmer  STEP 7:Define attributes hike, skills and methods display() & getDetails() STEP 8: In main class, create 3 objects for 3 programmers  STEP 9:Display output  STEP 10:Stop |
| **Program:** | **package** enheritence;  **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); |

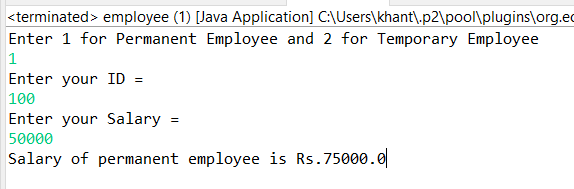
|  |  |
| --- | --- |
|  | System.***out***.println("Address : "+address);  }  **void** getDetails()  {  System.***out***.println("Enter name : "); name = in.nextLine(); System.***out***.println("Enter address : "); address = in.nextLine(); System.***out***.println("Enter age : "); age = in.nextInt();  }  }  **class** Employee **extends** Person{ **int** empID; **double** salary; Employee()  {  empID = 0;  salary = 0.0;  }  **void** getDetails()  {  **super**.getDetails(); System.***out***.println("Enter Employee ID : "); empID = in.nextInt(); System.***out***.println("Enter base salary : "); salary = in.nextDouble();  }  **void** display()  {  **super**.display(); System.***out***.println("Employee ID : "+empID);  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); |
| } |  |
| **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(); |
| } |  |
| } |  |
| **Input given:** | Enter details for 1st programmer Enter name : | |
| Raks |  |
| Enter | address : |
| abc |  |
| Enter | age : |
| 12 |  |
| Enter | Employee ID : |
| 11 |  |
| Enter | base salary : |
| 1000 |  |
| Enter | salary hike : |
| 12 |  |
| Enter technical skills : | |
| none |  |
| Enter details for 2nd programmer | |
| Enter | name : |
| ram |  |
| Enter | address : |
| abcd |  |
| Enter | age : |
| 23 |  |
| Enter | Employee ID : |
| 12 |  |
| Enter | base salary : |
| 1000 |  |
| Enter | salary hike : |

|  |  |
| --- | --- |
|  | 12  Enter technical skills : none  Enter details for 3rd programmer Enter name :  raj  Enter address :  asbd  Enter age :  33  Enter Employee ID : 13  Enter base salary : 1000  Enter salary hike : 12  Enter technical skills : none |
| **Output Screenshot:** |  |

# 2.

|  |  |
| --- | --- |
| **Algorithm**  **:** | STEP 1: Start  STEP 2: create class employee1, define attributes and methods setdetails() STEP 3: create child classes PermanentEmp and TemperoryEmp  STEP 4:define attributes and method generatesalary() in both the classes STEP 5:Create main function  STEP 6:Give the user 2 choices of permanent or temporary employee STEP 7:create object in main function according to the case selected STEP 8: print the output  STEP 9: Stop |
| **Program:** | **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** TemperoryEmp **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"); |
| **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"); |
| } |
| } |
| } |
| **Input** | Enter 1 for Permanent Employee and 2 for Temporary Employee |
| **given:** | 1  Enter your ID = |
|  | 100 |
|  | Enter your Salary = |
|  | 50000 |
| **Output** |  |
| **Screensho** |
| **t:** |