# Assignment:3

## Problem Statement: -

Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance wherever applicable. Employee class with Emp\_name, Emp\_id, Address, Mail\_id, and Mobile\_no as members. Inherit the classes, Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

## Objectives:

1. To Study Inheritance and its types
2. To implement inheritance using OOP language

## Theory:-

Inheritance:

Different kinds of objects often have a certain amount in common with each other. Mountain bikes, road bikes, and tandem bikes, for example, all share the characteristics of bicycles (current speed, current pedal cadence, current gear). Yet each also defines additional features that make them different: tandem bicycles have two seats and two sets of handlebars; road bikes have drop handlebars; some mountain bikes have an additional chain ring, giving them a lower gear ratio. Object-oriented programming allows classes to inherit commonly used state and behavior from other classes. In this example, Bicycle now becomes the superclass of MountainBike, RoadBike, and TandemBike. In the Java programming language, each class is allowed to have one direct superclass, and each superclass has the potential for an unlimited number of subclasses:

The syntax for creating a subclass is simple. At the beginning of your class declaration, use the extends keyword, followed by the name of the class to inherit from:

class MountainBike **extends** Bicycle {

// new fields and methods defining

// a mountain bike would go here

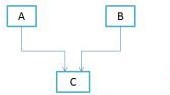
}

This gives MountainBike all the same fields and methods as Bicycle, yet allows its code to focus exclusively on the features that make it unique. This makes code for your subclasses easy to read. However, you must take care to properly document the state and behavior that each superclass defines, since that code will not appear in the source file of each subclass.

* **Single Inheritance:** When a class extends another one class only then we call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Here A is a parent class of B and B would be a child class of A.



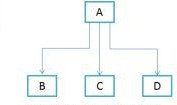
* Multiple Inheritance: It refers to the concept of one class extending (Or inherits) more than one base class. The inheritance we learnt earlier had the concept of one base class or parent. The problem with “multiple inheritance” is that the derived class will have to manage the dependency on two base classes.



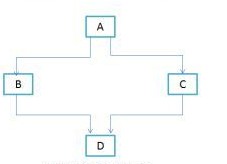
* Multilevel Inheritance: Multilevel inheritance refers to a mechanism in OO technology where one can inherit from a derived class, thereby making this derived class the base class for the new class. As you can see in below flow diagram C is subclass or child class of B and B is a child class of A.



* Hierarchical Inheritance: In such kind of inheritance one class is inherited by many sub classes. In below example class B,C and D inherits the same class A. A is parent class (or base class) of B,C & D.



* Hybrid Inheritance :In simple terms you can say that Hybrid inheritance is a combination of Single and Multiple inheritance. A typical flow diagram would look like below. A hybrid inheritance can be achieved in the java in a same way as multiple inheritance can be!! Using interfaces. yes you heard it right. By using interfaces you can have multiple as well as hybrid inheritance in Java.



## Steps :

1. Start
2. Create the class Employee with name, Empid, address, mailid, mobileno as data members.
3. Inherit the classes Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class.
4. Add Basic Pay (BP) as the member of all the inherited classes.
5. Calculate DA as 97% of BP, HRA as 10% of BP, PF as 12% of BP, Staff club fund as 0.1% of BP.
6. Calculate gross salary and net salary.
7. Generate payslip for all categories of employees.
8. Create the objects for the inherited classes and invoke the necessary methods to display the Payslip
9. Stop

## Input:

Empid, address, mailid, mobileno, Basic Pay (BP)

## Output:

gross and net salary slip

## Implementation: -

class Employee { int empid;

long mobile;

String name, address, mailid; void getdata() { }

void display() { }

}

class Programmer extends Employee { double salary,bp,da,hra,pf,club,net,gross; void getasst() { }

void calculateasst() { }

}

class TeamLead extends Employee {

---

}

class AssistantProjectManager extends Employee {

---

}

class Project Manager extends Employee {

---

}

class Salary {

public static void main(String args[]) { }

}

**Algorithm:-**

1. **Create a class employee and take the basic info input from the user about the employee.**
2. **Create class Programmer, Team lead, APM, Project Manager and use inheritance to link it to the employee class.**
3. **Ask for the basic pay in the respective classes .**
4. **Create a main class Inheritance and ask for option / choice to the user about the information he/she wants to enter .**
5. **Use switch case system to call all the defined functions.**

**Code : -**

package com.company;  
  
import java.util.Scanner;  
  
//public class Inheritance {  
 class employee{  
 Scanner sc = new Scanner(System.*in*);  
 String name,add,mail;  
 float id ,mobile,basic;  
  
 void getdata(){  
 System.*out*.println("Enter name of employee : ");  
 name = sc.nextLine();  
 System.*out*.println("Enter mail id of employee : ");  
 mail = sc.next();  
 System.*out*.println("Enter address of the employee : ");  
 add = sc.next();  
 System.*out*.println("Enter mobile number of employee : ");  
 mobile = sc.nextInt();  
  
 }  
  
 void display(){  
 System.*out*.println("\nEmp name :"+name);  
 System.*out*.println("\nEmp mail id :"+mail);  
 System.*out*.println("\nEmp address :"+add);  
 System.*out*.println("\nEmp mobile number :"+mobile);  
 }  
  
 void salary(){  
 float da , hra , cf , pf , gross;  
 da=basic\*97/100;  
 hra=basic\*10/100;  
 pf=basic\*12/100;  
 cf=basic\*0.1f/100;  
 gross=basic+da+hra-pf-cf;  
 System.*out*.println("Basic salary of employee : "+basic);  
 System.*out*.println("Gross salary of employee :"+gross);  
  
 }  
 }  
  
 class Programmer extends employee{  
 float salary;  
 void getprogrammer(){  
 System.*out*.print("Enter the basic pay: ");  
 basic=sc.nextFloat();  
 }  
 }  
 class APM extends employee{  
 float salary;  
 void getass(){  
 System.*out*.print("Enter basic pay : ");  
 basic=sc.nextFloat();  
 }  
 }  
 class TeamLead extends employee{  
 float salary;  
 void getteam(){  
 System.*out*.println("Enter basic pay : ");  
 basic=sc.nextFloat();  
 }  
 }  
 class ProjectManager extends employee{  
 float salary;  
 void getmanager(){  
 System.*out*.println("Enter basic pay : ");  
 basic=sc.nextFloat();  
 }  
 }  
 public class Inheritance{  
 public static void main(String[] args) {  
 int choice=0;  
 do{  
 System.*out*.println("Enter from the following : ");  
 System.*out*.println("\n1)Programmer"+"\n2)Assistant Project Manager"+"\n3)Team Lead"+"\n4)Project Manager");  
 Scanner c=new Scanner (System.*in*);  
 System.*out*.print("Enter your choice: ");  
 choice=c.nextInt();  
 switch(choice){  
 case 1:  
 Programmer p=new Programmer();  
 p.getdata();  
 p.getprogrammer();  
 p.display();  
 p.salary();  
 break;  
 case 2:  
 APM ass=new APM();  
 ass.getdata();  
 ass.getass();  
 ass.display();  
 ass.salary();  
 break;  
 case 3:  
 TeamLead tl=new TeamLead();  
 tl.getdata();  
 tl.getteam();  
 tl.display();  
 tl.salary();  
 case 4:  
 ProjectManager s=new ProjectManager();  
 s.getdata();  
 s.getmanager();  
 s.display();  
 s.salary();  
  
// default:  
// System.out.println("Enter correct choice number !");  
 }  
 }while(choice==1);  
 }  
 }

**Output :-**

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.1\lib\idea\_rt.jar=58008:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Rishabh\IdeaProjects\CollegeJavaAssignments\out\production\Assignment2 com.company.Inheritance

Enter from the following :

1)Programmer

2)Assistant Project Manager

3)Team Lead

4)Project Manager

Enter your choice: 1

Enter name of employee : Rishabh

Enter mail id of employee : jskfg

Enter address of the employee : sdjgls

Enter mobile number of employee : 45464

Enter the basic pay: 1200000

Emp name :Rishabh

Emp mail id :jskfg

Emp address :sdjgls

Emp mobile number :45464.0

Basic salary of employee : 1200000.0

Gross salary of employee :2338800.0

Enter from the following :

1)Programmer

2)Assistant Project Manager

3)Team Lead

4)Project Manager

Enter your choice: 2

Enter name of employee : sjshgw

Enter mail id of employee : vsldkj

Enter address of the employee : sdfkjsld

Enter mobile number of employee : 34634

Enter basic pay : 122343

Emp name :sjshgw

Emp mail id :vsldkj

Emp address :sdfkjsld

Emp mobile number :34634.0

Basic salary of employee : 122343.0

Gross salary of employee :238446.52

**Conclusion:-**

**Thus we have successfully executed the code for inheritance with output.**