**Ex.No. 3 PROGRAM TO GENERATE PAYSLIP USING INHERITANCE**

**Date:**

**1.0 Aim**

To develop a java application to generate pay slip for different category of employees using the concept of inheritance.

Develop a java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor 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

**2.0 Procedure:**

**STEP 1:** Create the specified classes

**STEP 2**: Add members of the inherited class

**STEP 3**: Calculate gross salary and net salary with the given equation

**STEP 4**: Display the pay slip with net salary and gross salary

**3.0 Program:**

package employee;

import java.io.IOException;

import java.util.Scanner;

class Emp

{

String ename,Address,email;

int eid;

int mobile;

void getEmployeedetails()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter the Emp\_id. :");

eid=in.nextInt();

System.out.println("Enter the Employee Name:");

ename=in.next();

System.out.println("Enter the Employee Address:");

Address=in.next();

System.out.println("Enter the Employee Email id :");

email=in.next();

System.out.println("Enter the Mobile No:");

mobile=in.nextInt();

}

void pay\_calulation(double BasicPay)

{

double DA,HRA,PF,Sfund,Gross\_Salary,Netsalary;

DA=BasicPay\*0.97;

HRA=BasicPay\*0.10;

PF=BasicPay\*0.12;

Sfund=BasicPay\*0.1;

Gross\_Salary=BasicPay+DA+HRA;

Netsalary=Gross\_Salary-(PF+Sfund);

System.out.println("Gross salary of the Employee"+Gross\_Salary);

System.out.println("Net salary of the Employee: "+Netsalary);

}

void display()

{

System.out.println("Emp\_id:"+eid);

System.out.println("Employee Name:"+ename);

System.out.println("Employee Address:"+Address);

System.out.println("Employee Email id :"+email);

System.out.println("Employee Mobile No:"+mobile);

}

}

class Programmer extends Emp

{

double BasicPay;

void Programmerdetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the Programmer:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class AssistantProfessor extends Emp

{

void APDetails()

{

double BasicPay;

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the AssistantProfessor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class AssociateProfessor extends Emp

{

double BasicPay;

void ASPDetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the AssociateProfessor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class Professor extends Emp

{

double BasicPay;

void profDetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the Professor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

public class Employee

{

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.println("Choose the type Employee");

System.out.println("1.Programmer ,2.Assistant Professor,3.Associate Professor ,4.Professor: ");

int ch=in.nextInt();

switch(ch)

{

case 1: System.out.println("PROGRAMMER DETAILS");

Programmer p=new Programmer();

p.Programmerdetails();

break;

case 2: System.out.println("Assistant Professor DETAILS");

AssistantProfessor ap=new AssistantProfessor();

ap.APDetails();

break;

case 3: System.out.println("Associate Professor DETAILS");

AssociateProfessor asp=new AssociateProfessor();

asp.ASPDetails();

break;

case 4: System.out.println("Professor DETAILS");

Professor pf=new Professor();

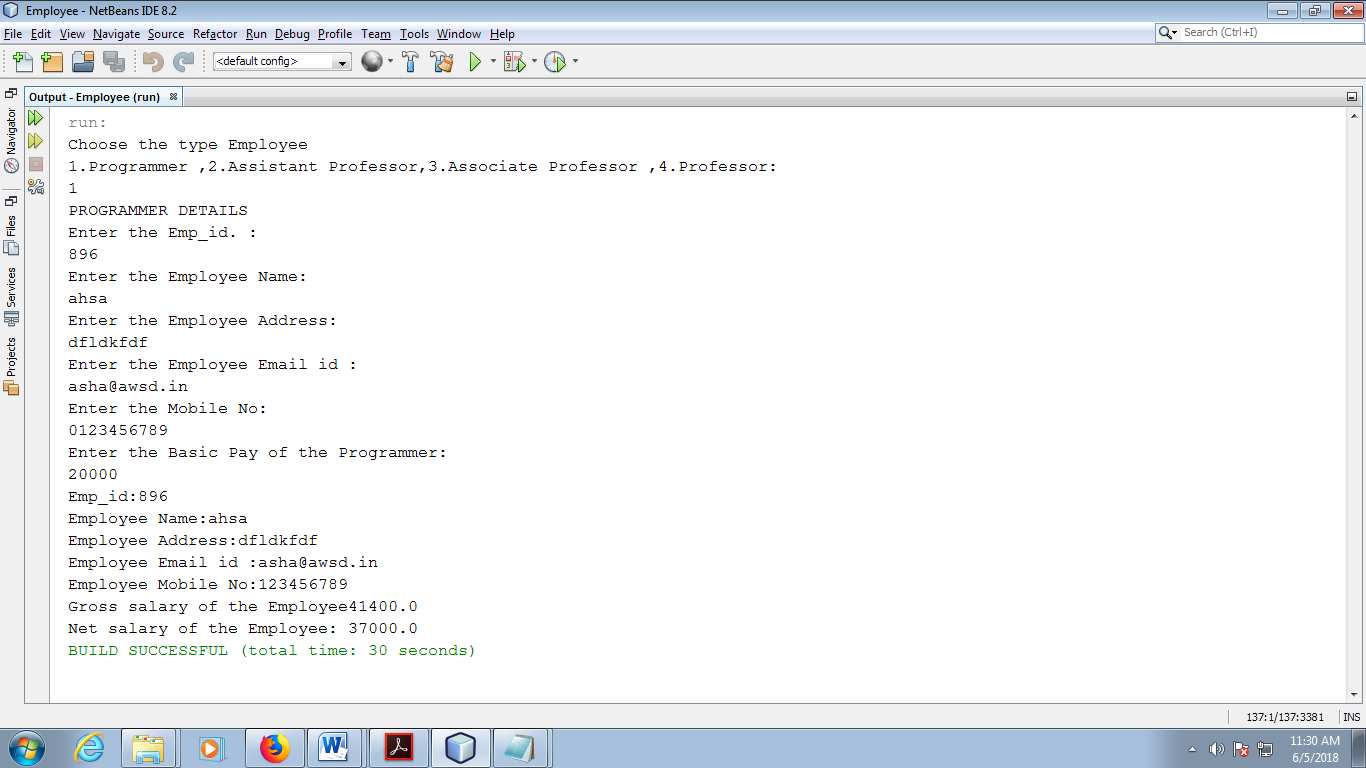
pf.profDetails();

break;

}

}}

**OUTPUT:**



**4.0 Result**:

Thus above program executed and output is verified.