**1.Create a class Animal with an abstract method numberOfLegs():int.The classes Dog,Human inherits from Animal.Dog has 4legs,Human has 2.**

**create 2 references of Animal and assign them the objects of Dog and Human invoke the method.**

**//Animal class**

**package** p1;

**public** **abstract** **class** Animal {

**public** **abstract** **int** numberOfLegs();

}

**//Dog class**

**package** p1;

**public** **class** Dog **extends** Animal{

**int** dog;

@Override

**public** **int** numberOfLegs() {

dog=4;

**return** dog;

}

}

**//Human class**

**package** p1;

**public** **class** HUman **extends** Animal{

**int** human;

@Override

**public** **int** numberOfLegs() {

**return** human=2;

}

}

**//Main**

**package** p1;

**public** **class** Demo {

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

Animal a;

a=**new** Dog();

System.***out***.println("The number of legs of dog is "+a.numberOfLegs());

a=**new** HUman();

System.***out***.println("The number of legs of human is "+a.numberOfLegs());

}

}

2.Develop an abstract class Employe with ID,name and address as arguments and an abstract method computeSalary().double.

The classes ContractEmployee and RegularEmploye inherit Person.ContactEmployee has hours Worked and hourly Wage as its attributes and the

RegularEmployee has basic and allowance as attributes.The rules for the computation of salary are as follows

1.For Contract employees the salary is product of hoursWorked and hourly Wage.

2.For regular employee the salary is basic + HRA(20%of basic)+allowance.

write the demo class with main() method to create objects of both ContractEmployee and RegularEmployee and compute the salary.Draw the class diagram.

//Employee class

**package** p2;

**public** **abstract** **class** Employee {

**long** Id;

String name;

String address;

**abstract** **double** computeSalary();

}

//Regular Employee

**package** p2;

**public** **class** RegularEmployee **extends** Employee{

**double** basicpay;

**double** allowance;

**double** sal;

RegularEmployee(**double** basicpay,**double** allowance)

{

**this**.basicpay=basicpay;

**this**.allowance=allowance;

}

@Override

**double** computeSalary() {

**double** HRA=20/100\*basicpay;

sal=basicpay+HRA+allowance;

System.***out***.println("SALARY FOR REGULAR EMPLOYEE IS RS "+sal);

**return** sal;

}

}

//Contract employee

**package** p2;

**public** **class** ContractEmpployee **extends** Employee {

**double** hourworked;

**double** wage;

ContractEmpployee(**double** worked,**double** wage)

{

hourworked=worked;

**this**.wage=wage;

}

@Override

**double** computeSalary() {

**double** sal;

sal=hourworked\*wage;

System.***out***.println("SALARY FOR CONTRACT EMPLOYEE IS RS "+sal);

**return** sal;

}

}

//Employee Demo

**package** p2;

**public** **class** EmployeeDemo {

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

Employee e;

e=**new** RegularEmployee(30000,10000);

e.computeSalary();

e=**new** ContractEmpployee(9,5000);

e.computeSalary();

}

}

3.Develop an interface Game with a method noOfplayers():int.The classes Chess and FootBall implement Game. Chess has 2 players and

FootBall has 11 players. Create objects of Chess,FootBall and invoke the overridden method.

//Game

**package** p3;

**interface** Game {

**public** **abstract** **int** noOfplayers();

}

//Chess

**package** p3;

**public** **class** Chess **implements** Game{

@Override

**public** **int** noOfplayers() {

**return** 2;

}

}

//Football

**package** p3;

**public** **class** Fotball **implements** Game{

@Override

**public** **int** noOfplayers() {

**return** 11;

}

}

//main

**package** p3;

**public** **class** Demo {

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

Game g;

g=**new** Chess();

System.***out***.println("Players in chess is "+g.noOfplayers());

g=**new** Fotball();

System.***out***.println("Players in fotball is "+g.noOfplayers());

}

}