**EXERCISE 3**

**1) 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**

***class* Employee{**

**String Empennage, Emp\_id,Address, Mail\_id;**

**long Mob\_no;**

**Employee(String name,String id,String addr,String mail,long phn){**

**Emp\_name=name;**

**Emp\_id=id;**

**Address=addr;**

**Mail\_id=mail;**

**Mob\_no=phn;**

**}**

**}**

**class Assistant\_professor extends Employee{**

**int bp;**

**Assistant\_professor(String name,String id,String addr,String mail,long phn,int pay){**

**super(name,id,addr,mail,phn);**

**bp = pay;**

**}**

**void payslip(){**

**float da = bp\*(0.97f);**

**float hra = bp\*(0.10f);**

**float pf = bp\*(0.12f);**

**float sc = bp\*(0.001f);**

**float net\_salary = (da + hra)- (pf+sc);**

**float gross\_salary = (da+hra);**

**System.out.println("the gross salary is"+ gross\_salary);**

**System.out.println("the net salary is"+ net\_salary);**

**}**

**}**

**class Associate\_professor extends Employee{**

**int bp;**

**Associate\_professor(String name,String id,String addr,String mail,long phn,int pay){**

**super(name,id,addr,mail,phn);**

**bp = pay;**

**}**

**void payslip(){**

**float da = bp\*(0.97f);**

**float hra = bp\*(0.10f);**

**float pf = bp\*(0.12f);**

**float sc = bp\*(0.001f);**

**float net\_salary = (da + hra)- (pf+sc);**

**float gross\_salary = (da+hra);**

**System.out.println("the gross salary is"+ gross\_salary);**

**System.out.println("the net salary is"+ net\_salary);**

**}**

**}**

**class Professor extends Employee{**

**int bp;**

**Professor(String name,String id,String addr,String mail,int phn,int pay){**

**super(name,id,addr,mail,phn);**

**bp = pay;**

**}**

**void payslip(){**

**float da = bp\*(0.97f);**

**float hra = bp\*(0.10f);**

**float pf = bp\*(0.12f);**

**float sc = bp\*(0.001f);**

**float net\_salary = (da + hra)- (pf+sc);**

**float gross\_salary = (da+hra);**

**System.out.println("the gross salary is"+ gross\_salary);**

**System.out.println("the net salary is"+ net\_salary);**

**}**

**}**

**public class Main**

**{**

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

**Assistant\_professor ap = new Assistant\_professor("balasurya","4164","s","abc@gmail.com",9876543210,15600);**

**Associate\_professor ap1 = new Associate\_professor("harish","4000","s","xyz@gmail.com",1234567890,32000);**

**Professor p = new Professor("chocolate","4008","s","bala@gmail.com",9876543210,52000);**

**ap.payslip();**

**ap1.payslip();**

**p.payslip();**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2) Create a Circle class with following members.**

**A data member that stores the radius of a circle**

**A constructor function with an argument that initializes the radius**

**A function that computes and returns are of a circle**

**Create two derived classes Sector and Segment that inherit the Circle class.**

**Both classes inherit radius and the function that returns the circle&#39;s area from**

**Circle. In addition to the members inherited from Circle, Sector and Segment**

**have some specific members as follows:**

**: Sector**

**A data member that stores the control angle of a sector(in radians)**

**A constructor function with arguments that initialize radius and angle**

**A function that computes and returns the area of a sector**

**Segment**

**A data member that stores the angle of a segment in a circle**

**A constructor function with arguments that initialize radius and angle**

**A function that computes and returns the area of a segment**

**Create the main () function to instantiate an object of each class and then call**

**appropriate member**

**: Note :Area\_of\_circle =π r 2**

**Area\_of\_Sector=r 2 θ/2**

**Area\_of\_segment= ½ \* r 2 ( θ – sin θ)**

**class Circle{**

**float radius;**

**public Circle(float r){**

**radius = r;**

**}**

**float cirArea(){**

**return (3.14f\*radius\*radius);**

**}**

**}**

**class Sector extends Circle{**

**float angle;**

**public Sector(float r,float ang){**

**super(r);**

**angle = ang;**

**}**

**float secArea(){**

**return (0.5f\*(radius\*radius)\*angle);**

**}**

**}**

**class Segment extends Circle{**

**float angle;**

**public Segment(float r,float ang){**

**super(r);**

**angle = ang;**

**}**

**float segArea(){**

**return ((float)(0.5f\*(radius\*radius)\*(angle-Math.sin(angle))));**

**}**

**}**

**public class Main**

**{**

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

**Circle c = new Circle(5);**

**float cir = c.cirArea();**

**System.out.println("the circle area is "+ cir);**

**Sector s = new Sector(7,33);**

**float sec = s.secArea();**

**System.out.println("the sector area is "+ sec);**

**Segment sg = new Segment(17,33);**

**float seg = sg.segArea();**

**System.out.println("the segment area is "+ seg);**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3) Abstract program**

**abstract class Figure{**

**double dim1,dim2;**

**Figure(double a,double b){**

**dim1 = a;**

**dim2 = b;**

**}**

**abstract double area();**

**}**

**class Rectangle extends Figure{**

**Rectangle(double a,double b){**

**super(a,b);**

**}**

**double area(){**

**return dim1\*dim2;**

**}**

**}**

**class Triangle extends Figure{**

**Triangle(double a,double b){**

**super(a,b);**

**}**

**double area(){**

**return (0.5\*dim1\*dim2);**

**}**

**}**

**public class Main**

**{**

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

**Rectangle r = new Rectangle(7,3);**

**Triangle t= new Triangle(7,7);**

**Figure superref;**

**superref = r;**

**System.out.println("area is "+ superref.area());**

**superref = t;**

**System.out.println("area is "+ superref.area());**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_$\_\_\_\_\_\_\_\_\_\_\_**