Code Red Technologies

Grade settings: Maximum grade: 100
Run: Yes **Evaluate**: Yes
Automatic grade: Yes

Code Red Technologies is a software company. There are more than 10,000 employees in the company. The management wants to provide incremented salary to their employees according to the tier in which they work. The tier they work varies according to the years of experience. The company wants to calculate the incremented salary using their main computer. Assist them in calculating and retrieving the incremented salary of the employees according to the tier in which they work.

Requirement 1: Calculation of Incremented Employee Salary

The application needs to calculate the incremented salary to be paid to the Employees according to the tier in which they work.

Component Specification: Employee Class (Parent Class)

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
	Employee	String EmployeeId String EmployeeName int yearsOfExperience String gender double salary	Include a public getter and setter method for all the attributes. Include a public 5 argument constructor in the given order - EmployeeId, EmployeeName, yearsOfExperience, gender and salary.	
Calculation of incremented salary	Employee		public double calculateIncrementedSalary(int incrementPercentage)	This is an abstract method.

Note: The attributes of the Employee class should be protected and method and the constructor is declared public.

Component Specification: CasualEmployee class (Needs to be a child of the Employee class)

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
	CasualEmployee	int supplementaryHours double foodAllowance	Include necessary setters and getters and a public 7 argument constructor in the given order - EmployeeId, EmployeeName, yearsOfExperience, gender, salary, supplementaryHours, and foodAllowance.	
Calculation of Incremented salary	CasualEmployee		public double calculateIncrementedSalary(int incrementPercentage)	This method should calculate the salary according to the increment percentage and return the same.

Note: The attributes of the CasualEmployee class should be private and the methods and constructor should be declared public.

Calculation of Incremented Salary for Casual Employee:

For each supplementary hour, employee earns Rs 1000.

So, total salary = (supplementaryHours *1000)+foodAllowance+current salary

incrementedSalary = totalSalary+(totalSalary*incrementPercentage/100)

Component Specification: TraineeEmployees class (Needs to be a child of the Employee class)

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
	TraineeEmployees	int supplementaryTrainingHours int scorePoints	Include necessary setters and getters and a public 7 argument constructor in the given order - EmployeeId, EmployeeName, yearsOfExperience, gender, salary,	

			supplementaryTrainingHours and scorePoints.	
Calculation of Incremented salary	TraineeEmployees		public double calculateIncrementedSalary(int incrementPercentage)	This method should calculate the salary according to the increment percentage and return the same.

Note: The attributes of the TraineeEmployees class should be private and methods and constructor should be declared as public.

Calculation of Incremented Salary for Trainee Employee

For each supplementary training hour, employee earns Rs 500.

For each score point, employee earns Rs 50.

So, total salary = (supplementaryTrainingHours *500)+(scorePoints*50)+current salary

incrementedSalary = totalSalary+(totalSalary*incrementPercentage/100)

Component Specification: PermanentEmployee class (Needs to be a child of the Employee class)

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
	PermanentEmployee	double medicalAllowance double VehicleAllowance	Include necessary setters and getters and a public 7 argument constructor in the given order - EmployeeId, EmployeeName, yearsOfExperience, gender, salary, medicalAllowance and VehicleAllowance.	
Calculation of Incremented salary	PermanentEmployee		public double calculateIncrementedSalary(int incrementPercentage)	This method should calculate the salary according to the increment

				percentage and return the same.

Note: The attributes of the PermanentEmployee class should be private and the methods and constructors should be declared as public.

Calculation of Incremented Salary for Permanent Employee

total salary = medicalAllowance+VehicleAllowance+current salary

incrementedSalary = totalSalary+(totalSalary*incrementPercentage/100)

Use a **public class UserInterface** with the main method to test the application.

- Get the inputs from the user as mentioned in the sample input.
- If the year of experience is between 1 and 5(inclusive) they fall under the category of **trainee employee** and if the year of experience is between 6 and 10(inclusive) they fall under **casual employee** if the year of experience is between 11 and 25(inclusive) they fall under **senior employee**. If the year of experience is out of these ranges listed above, print as **"Provide valid Years of Experience"**.
- Based on the category of employee, call the corresponding constructor to set the values to the Employee object and then call calculateIncrementedSalary() method with increment percentage as argument to calculate the increment salary amount.

Assumptions:

- **The increment percentage for Trainee Employee is 5.** Pass the value 5 to the method public double calculateIncrementedSalary(int incrementPercentage) present in TraineeEmployees class
- **The increment percentage for Casual Employee is 12.** Pass the value 12 to the method public double calculateIncrementedSalary(int incrementPercentage) present in CasualEmployee class
- **The increment percentage for Permanent Employee is 12**. Pass the value 12 to the method public double calculateIncrementedSalary(int incrementPercentage) present in PermanentEmployee class

Note:

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes and methods as specified in the question.

- Adhere to the code template, if provided.

Sample Input 1:

Enter Employee Id

GHJ76

Enter Employee name

Jack

Enter Experience in years

4

Enter Gender

Male

Enter Salary

32000

Enter Supplementary Training Hours

9

Enter Score Points

8

Incremented Salary is 38745.0

Sample Input 2:

Enter Employee Id

ASM890

Enter Employee name

Benny

Enter Experience in years

8

Enter Gender

Male

Enter Salary

50000

Enter Supplementary Hours

15

Enter Food Allowance

2700

Incremented Salary is 75824.0

Sample Input 3:

Enter Employee Id

CVB621

Enter Employee name

Malini

Enter Experience in years

18

Enter Gender

Female

Enter Salary

67000

Enter Medical Allowance

1800

Enter Vehicle Allowance

2300

Incremented Salary is 79632.0

Sample Input 4:

Enter Employee Id

MCH486

Enter Employee name

Mano

Enter Experience in years

89

Enter Gender

Male

Enter Salary

78000

Provide valid Years of Experience

Solution:

CasualEmployee.java

```

public class CasualEmployee extends Employee{

	private int supplementaryHours;
private double foodAllowance;

public int getSupplementaryHours() {
        return supplementaryHours;
}
public void setSupplementaryHours(int supplementaryHours) {
        this.supplementaryHours = supplementaryHours;
}
public double getFoodAllowance() {
        return foodAllowance;
}
public void setFoodAllowance(double foodAllowance) {
        this.foodAllowance = foodAllowance;
}
public CasualEmployee(String EmployeeId, String EmployeeName, int yearsOfExperience, String gender, double salary, int supplementaryHours, double foodAllowance)
{
  super(EmployeeId, EmployeeName,yearsOfExperience,gender,salary);
  this.supplementaryHours=supplementaryHours;
  this.foodAllowance=foodAllowance;
}

public double calculateIncrementedSalary(int incrementPercentage)
{
        double total =(supplementaryHours*1000)+foodAllowance+this.salary;
        double incsalary=total+(total*incrementPercentage/100);
```

```
                return incsalary;
        }

}

```

Employee.java

```

public abstract class Employee {

        protected String EmployeeId;
        protected String EmployeeName;
        protected int yearsOfExperience;
        protected String gender;
        protected double salary;

        public abstract double calculateIncrementedSalary(int incrementPercentage);

        public String getEmployeeId() {
                return EmployeeId;
        }
        public void setEmployeeId(String employeeId) {
                this.EmployeeId = employeeId;
        }
        public String getEmployeeName() {
                return EmployeeName;
        }
```

```
public void setEmployeeName(String employeeName) {
        this.EmployeeName = employeeName;
}
public int getYearsOfExperience() {
        return yearsOfExperience;
}
public void setYearsOfExperience(int yearsOfExperience) {
        this.yearsOfExperience = yearsOfExperience;
}
public String getGender() {
        return gender;
}
public void setGender(String gender) {
        this.gender = gender;
}
public double getSalary() {
        return salary;
}
public void setSalary(double salary) {
        this.salary = salary;
}
public Employee(String employeeId, String employeeName, int yearsOfExperience, String
gender, double salary) {
        super();
        this.EmployeeId = employeeId;
        this.EmployeeName = employeeName;
        this.yearsOfExperience = yearsOfExperience;
        this.gender = gender;
        this.salary=salary;
    }
}

```

PermanentEmployee.java

```

public class PermanentEmployee extends Employee{

    private double medicalAllowance;
    private double VehicleAllowance;

    public double getMedicalAllowance() {
        return medicalAllowance;
    }

    public void setMedicalAllowance(double medicalAllowance) {
        this.medicalAllowance = medicalAllowance;
    }

    public double getVehicleAllowance() {
        return VehicleAllowance;
    }

    public void setVehicleAllowance(double vehicleAllowance) {
```

```java
            VehicleAllowance = vehicleAllowance;
    }

    public PermanentEmployee(String EmployeeId, String EmployeeName, int yearsOfExperience,
String gender, double salary, double medicalAllowance, double vehicleAllowance)
    {
      super(EmployeeId, EmployeeName, yearsOfExperience, gender, salary);
      this.medicalAllowance=medicalAllowance;
      this.VehicleAllowance=vehicleAllowance;
    }

    public double calculateIncrementedSalary(int incrementPercentage)
    {
            double total=medicalAllowance + VehicleAllowance+this.salary;
            double incsalary=total+(total*incrementPercentage/100);
            return incsalary;
    }
}

```

TraineeEmployees.java

```java

public class TraineeEmployees extends Employee{


    private int supplementaryTrainingHours;
    private int scorePoints;
```

```

public int getSupplementaryTrainingHours() {
        return supplementaryTrainingHours;
}
public void setSupplementaryTrainingHours(int supplementaryTrainingHours) {
        this.supplementaryTrainingHours = supplementaryTrainingHours;
}
public int getScorePoints() {
        return scorePoints;
}
public void setScorePoints(int scorePoints) {
        this.scorePoints = scorePoints;
}

public TraineeEmployees(String EmployeeId, String EmployeeName, int yearsOfExperience, String gender, double salary, int supplementaryTrainingHours, int scorePoints)
{
   super(EmployeeId, EmployeeName, yearsOfExperience, gender, salary);
   this.supplementaryTrainingHours=supplementaryTrainingHours;
   this.scorePoints=scorePoints;
}

public double calculateIncrementedSalary(int incrementPercentage){
        double total=(supplementaryTrainingHours*500)+(scorePoints*50)+this.salary;
        double incsalary=total+(total*incrementPercentage/100);
        return incsalary;
```

```
    }



}

```

UserInterface.java

```
import java.util.Scanner;
public class UserInterface {

    public static void main(String[] args){

            Scanner sc=new Scanner(System.in);
            System.out.println("Enter Employee Id");
            String EmployeeId = sc.next();
            System.out.println("Enter Employee name");
            String EmployeeName = sc.next();
            System.out.println("Enter Experience in years");
            int yearsOfExperience = sc.nextInt();
            System.out.println("Enter Gender");
            String gender = sc.next();
      System.out.println("Enter Salary");
      double salary=sc.nextDouble();
      double incSalary=0;
      if(yearsOfExperience>=1 && yearsOfExperience <= 5)
        {
          System.out.println("Enter Supplementary Training Hours");
```

```
            int supplementaryTrainingHours = sc.nextInt();
            System.out.println("Enter Score Points");
            int scorePoints = sc.nextInt();
            TraineeEmployees te=new TraineeEmployees(EmployeeId, EmployeeName, yearsOfExperience, gender, salary, supplementaryTrainingHours, scorePoints);
            incSalary=te.calculateIncrementedSalary(5);
            System.out.println("Incremented Salary is "+incSalary);
        }
        else if(yearsOfExperience>=6 && yearsOfExperience <=10)
        {
            System.out.println("Enter Supplementary Hours");
            int supplementaryHours = sc.nextInt();
            System.out.println("Enter Food Allowance");
            double foodAllowance = sc.nextDouble();
            CasualEmployee ce=new CasualEmployee(EmployeeId, EmployeeName, yearsOfExperience, gender, salary, supplementaryHours, foodAllowance);
            incSalary = ce.calculateIncrementedSalary(12);
            System.out.println("Incremented Salary is "+incSalary);
        }
        else if(yearsOfExperience>=10 && yearsOfExperience <=25)
        {
            System.out.println("Enter Medical Allowance");
            double medicalAllowance = sc.nextDouble();
            System.out.println("Enter Vehicle Allowance");
            double vehicleAllowance = sc.nextDouble();
            PermanentEmployee pe = new PermanentEmployee(EmployeeId, EmployeeName, yearsOfExperience, gender, salary, medicalAllowance, vehicleAllowance);
            incSalary=pe.calculateIncrementedSalary(12);
            System.out.println("Incremented Salary is "+incSalary);
        }
        else
            System.out.println("Provide valid Years of Experience");
    }

}

```