Code RED

CasualEmployee:

```
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:

```
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:

```
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) {
 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;
}
}
```

TraineeEmployee:

```
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:

```
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");

}  }
```