ASSIGNMENT3 Sandeep Sir

1.Loan Amortization Calculator

Implement a system to calculate and display the monthly payments for a mortgage loan. The system should:

Accept the principal amount (loan amount), annual interest rate, and loan term (in years) from the user.

Calculate the monthly payment using the standard mortgage formula:
o Monthly Payment Calculation:
▪ monthlyPayment = principal * (monthlyInterestRate * (1 + monthlyInterestRate)^(numberOfMonths)) / ((1 + monthlyInterestRate)^(numberOfMonths) - 1)
▪ Where monthlyInterestRate = annualInterestRate / 12 / 100 and numberOfMonths = loanTerm * 12
▪ Note: Here ^ means power and to find it you can use Math.pow( ) method

Display the monthly payment and the total amount paid over the life of the loan, in Indian Rupees (₹).
Define class LoanAmortizationCalculator with methods acceptRecord, calculateMonthlyPayment & printRecord and test the functionality in main method.

```
package org.example;

import java.util.Scanner;

 class LoanAmortizationCalculator{
    private double principal;
    private double annualInterestRate;
    private int loanTerm;

    void acceptRecord() {
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Principle:");
        principal=sc.nextDouble();
        System.out.println("anual interest:");
        annualInterestRate=sc.nextDouble();
        System.out.println("LoanT:");
        loanTerm=sc.nextInt();
    }

    public double calculateMonthlyPayment() {
        double monthlyInterestRate = (annualInterestRate / 12) /
        int numberOfMonths = loanTerm * 12;

        double monthlyPayment = principal * (monthlyInterestRate
        return monthlyPayment;
    }

    void printRecord () {
        double monthlyPayment=calculateMonthlyPayment();
        int numberOfMonths = loanTerm * 12;
        double totalAmount=monthlyPayment*numberOfMonths;
        System.out.println("monthlyPayment:₹"+monthlyPayment);
        System.out.println("total amount paid over the life :₹"+

    }
}

public class Program1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        LoanAmortizationCalculator loan=new LoanAmortizationCalc
       // loan.calculateMonthlyPayment();
        loan.acceptRecord();
        loan.printRecord();
    }
```

```
}
```

```
Principle:10000
anual interest:1.2
LoanT:2
monthlyPayment:₹421.8949551101263
total amount paid over the life :₹10125.478922643031
```

2.Compound Interest Calculator for Investment

Develop a system to compute the future value of an investment with compound interest. The system should:

Accept the initial investment amount, annual interest rate, number of times the interest is compounded per year, and investment duration (in years) from the user.

Calculate the future value of the investment using the formula:

- Future Value Calculation:
  - futureValue = principal * (1 + annualInterestRate / numberOfCompounds)^(numberOfCompounds * years)
- Total Interest Earned: totalInterest = futureValue - principal

Display the future value and the total interest earned, in Indian Rupees (₹).
Define class CompoundInterestCalculator with methods acceptRecord , calculateFutureValue, printRecord and test the functionality in main method.

```
package org.example;
import java.util.Scanner;
class CompoundInterestCalculator{
    private double investment_amount;
    private double annual_interest_rate;
    private int numberOfCompounds;
    private int investment_duration ;

     void acceptRecord() {
```

```
          Scanner sc=new Scanner(System.in);
        System.out.println("initial investment amount:");
        this.investment_amount=sc.nextDouble();
        System.out.println("annual interest rate:");
        this.annual_interest_rate=sc.nextDouble();
        System.out.println("number of times the interest is comp
        this.numberOfCompounds=sc.nextInt();
        System.out.println("investment duration :");
        this.investment_duration=sc.nextInt();

    }
    public double calculateFutureValue(){
        double futureValue = investment_amount * (1 + Math.pow(a
        return futureValue;
    }
     void printRecord() {
         double futureValue=calculateFutureValue();
         double totalInterest = futureValue - investment_amount
         System.out.println("Future Value:"+futureValue);
         System.out.println("Total interest eared:₹ "+totalInter

     }
}
public class Program2 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        CompoundInterestCalculator cmp=new CompoundInterestCalcu
        cmp.acceptRecord();
        cmp.calculateFutureValue();
        cmp.printRecord();

    }

}
```

```
Initial investment amount:10000
annual interest rate:1.2
number of times the interest is compounded per year:1
investment duration :1
Future Value:22000.0
Total interest eared:₹ 12000.0
```

3.BMI (Body Mass Index) Tracker

Create a system to calculate and classify Body Mass Index (BMI). The system should:

Accept weight (in kilograms) and height (in meters) from the user.

Calculate the BMI using the formula:

- BMI Calculation: BMI = weight / (height * height)

Classify the BMI into one of the following categories:

- Underweight: BMI < 18.5
- Normal weight: 18.5 ≤ BMI < 24.9
- Overweight: 25 ≤ BMI < 29.9
- Obese: BMI ≥ 30

Display the BMI value and its classification.

Define class BMITracker with methods acceptRecord, calculateBMI, classifyBMI & printRecord and test the functionality in main method.

```
package org.example;
import java.util.Scanner;
class BMITracker{
    private double weight;
    private double height;

    void acceptRecord() {
        Scanner sc=new Scanner(System.in);

        System.out.println("WEIGHT:");
        this.weight=sc.nextDouble();
```

```
        System.out.println("HEIGHT:");
        this.height=sc.nextDouble();
    }
    public double calculateBMI() {
        double BMI = weight / (height * height);
        return BMI;
    }
    void printRecord() {
        double BMI=calculateBMI();
        System.out.println("BMI:"+BMI);
        if(BMI<18.5) {
            System.out.println("Underweight");
        }
        else if(BMI>=18.5 && BMI < 24.9) {
            System.out.println("Normal weight");
        }
        else if(BMI>=25&&BMI<24.9) {
            System.out.println("Overweight");
        }
        else {
            System.out.println("Obese");
        }
    }

    }
public class Program3 {

    public static void main(String[] args) {
        BMITracker bmi=new BMITracker();
        bmi.acceptRecord();
        bmi.calculateBMI();
        bmi.printRecord();

    }
```

```
}
```

```
WEIGHT in kg:60
HEIGHT in m:2.1
BMI:13.605442176870747
Underweight
```

4.Discount Calculation for Retail Sales

Design a system to calculate the final price of an item after applying a discount. The system should:

Accept the original price of an item and the discount percentage from the user.

Calculate the discount amount and the final price using the following formulas:

- Discount Amount Calculation: discountAmount = originalPrice * (discountRate / 100)
- Final Price Calculation: finalPrice = originalPrice - discountAmount

Display the discount amount and the final price of the item, in Indian Rupees (₹). Define class DiscountCalculator with methods acceptRecord, calculateDiscount & printRecord and test the functionality in main method.

```
package org.example;

import java.util.Scanner;

class DiscountCalculator{
    private double originalPrice;
    private double discountRate;

    void acceptRecord() {
        Scanner sc=new Scanner(System.in);
        System.out.print("Original price:");
        this.originalPrice=sc.nextDouble();
```

```
        System.out.print("Discount rate::");
        this.discountRate=sc.nextDouble();

    }
    public double calculateDiscount() {
        double  discountAmount = originalPrice * (discountRate /
        return discountAmount;
    }
    void printRecord() {
        double  discountAmount=calculateDiscount();
        double finalPrice = originalPrice - discountAmount;
        System.out.println("Discount Amount:₹ "+discountAmount);
        System.out.println("Final Price:₹ "+finalPrice);
    }
}
public class Program4 {

    public static void main(String[] args) {
        DiscountCalculator cal=new DiscountCalculator();
        cal.acceptRecord();
        cal.calculateDiscount();
        cal.printRecord();

    }

}
```

```
Original price:10000
Discount rate::10
Discount Amount:₹ 1000.0
Final Price:₹ 9000.0
```

5.Toll Booth Revenue Management

Develop a system to simulate a toll booth for collecting revenue. The system

should:

1. Allow the user to set toll rates for different vehicle types: Car, Truck, and Motorcycle.
2. Accept the number of vehicles of each type passing through the toll booth.
3. Calculate the total revenue based on the toll rates and number of vehicles.
4. Display the total number of vehicles and the total revenue collected, in Indian Rupees (₹).
   - Toll Rate Examples:
   - Car: ₹50.00
   - Truck: ₹100.00
   - Motorcycle: ₹30.00

   Define class TollBoothRevenueManager with methods acceptRecord, setTollRates, calculateRevenue & printRecord and test the functionality in main method.

```
package org.example;

import java.util.Scanner;

class TollBoothRevenueManager{
    private double tollRateCar;
    private double tollRateTruck;
    private double tollRateMotorcycle;

     int totalCar;
     int totalTruck;
     int totalMotorcycle;


    void acceptRecord() {
        Scanner sc=new Scanner(System.in);
        System.out.print("totalCar:");
        this.totalCar=sc.nextInt();
```

```
        System.out.print("totalTruck:");
        this.totalTruck=sc.nextInt();
        System.out.print("totalMotorcycle:");
        this.totalMotorcycle=sc.nextInt();


    }
    public void setTollRates() {
        Scanner sc=new Scanner(System.in);
        System.out.print(" TOLL RATE FOR CAR:₹");
        this.tollRateCar=sc.nextDouble();
        System.out.print(" TOLL RATE FOR TRUCK:₹");
        this.tollRateTruck=sc.nextDouble();
        System.out.print(" TOLL RATE FOR MOTORCYLE:₹");
        this.tollRateMotorcycle=sc.nextDouble();
    }
    public double calculateRevenue() {
        double totalREvenue=(totalCar*tollRateCar)+(totalTruck*t
        return totalREvenue;
    }
    void printRecord() {
        double totalREvenue=calculateRevenue();
        System.out.println("Total Revenue:₹"+totalREvenue);
    }

}
public class Program5 {

    public static void main(String[] args) {
        TollBoothRevenueManager tbr=new TollBoothRevenueManager(
        tbr.acceptRecord();
        tbr.setTollRates();
        tbr.calculateRevenue();
        tbr.printRecord();

    }
```

```
}
```

```
totalCar:2
totalTruck:1
totalMotorcycle:3
 TOLL RATE FOR CAR:₹50
 TOLL RATE FOR TRUCK:₹100
 TOLL RATE FOR MOTORCYLE:₹50
Total Revenue:₹350.0
```