ASSIGNMENT NO.3

Note:

- The assignment is designed to practice class, fields, and methods only.
- Create a separate project for each question.
- Do not use getter/setter methods or constructors for these assignments.
- Define two classes: one class to implement the logic and another class to test it.

1. Loan Amortization Calculator

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

1. Accept the principal amount (loan amount), annual interest rate, and loan term (in years) from the user.
2. Calculate the monthly payment using the standard mortgage formula:
   - **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
3. 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.

Ans.

```
package org.example.ques1;



import java.util.Scanner;



class loanCalculator{

	double principal_amt;

	double annual_interest_rate;

	int loan_term;



	public void acceptRecord() {
```

```
                Scanner sc = new Scanner(System.in);

                System.out.println("Principal Amount : ");

                principal_amt = sc.nextDouble();

                System.out.println("Annual Interest Rate : ");

                annual_interest_rate = sc.nextDouble();

                System.out.println("Years : " );

                loan_term = sc.nextInt();

        }


        public double calculateMonthlyPayment(){

                double monthlyInterestRate = annual_interest_rate/12/100;

                int numberOFMonths = loan_term*12;

                double numerator = monthlyInterestRate * Math.pow(1+
monthlyInterestRate, numberOFMonths);

                double denominator = Math.pow(1+ monthlyInterestRate,
numberOFMonths) -1 ;

                return principal_amt * (numerator/denominator);

        }


        public void printRecord() {

                double monthlyPayment = calculateMonthlyPayment();

                double totalPayment = monthlyPayment*loan_term*12;



                 System.out.printf("Monthly Payment: ₹%.2f%n", monthlyPayment);

             System.out.printf("Total Amount Paid Over the Life of the Loan:
₹%.2f%n", totalPayment);
```

```
        }


    }


    public class question1 {


        public static void main(String[] args) {

            loanCalculator loan = new loanCalculator();


            loan.acceptRecord();


            loan.printRecord();




        }


    }
```

2. Compound Interest Calculator for Investment

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

1. 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.
2. 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`

3. 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.

Ans.

```
package org.example.ques2;

import java.util.Scanner;

class compoundInterest{
    int amount;
    double annualInterestRate;
    int numberOfCompounds;
    int years;

    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);
        System.out.println("amount :");
        amount = sc.nextInt();
        System.out.println("annual interest rate :");
        annualInterestRate = sc.nextDouble();
        System.out.println("Number of compounds :");
        numberOfCompounds = sc.nextInt();
        System.out.println("Years(inverstment Duration) :");
        years = sc.nextInt();
    }

    public double futureValueCalculation() {
         double ratePerPeriod = annualInterestRate / 100 / numberOfCompounds;
       double exponent = numberOfCompounds * years;
       return amount * Math.pow(1 + ratePerPeriod, exponent);
    }

    public void printRecord() {
        double futureValue = futureValueCalculation();
   double totalInterest = futureValue - amount;

   System.out.printf("Future Value of the Investment: ₹%.2f%n", futureValue);
   System.out.printf("Total Interest Earned: ₹%.2f%n", totalInterest);
    }

}


public class question2 {

    public static void main(String[] args) {
        compoundInterest cmpd  = new compoundInterest();
```

```
                cmpd.acceptRecord();

                cmpd.printRecord();

        }

}
```

3. BMI (Body Mass Index) Tracker

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

1. Accept weight (in kilograms) and height (in meters) from the user.
2. Calculate the BMI using the formula:
   - **BMI Calculation:** `BMI = weight / (height * height)`
3. 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
4. Display the BMI value and its classification.

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

Ans.

```
package org.example.ques3;
import java.util.Scanner;

class bmi{
        int weight;
        double height;

        public void acceptRecord() {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter Weight : ");
                weight = sc.nextInt();
                System.out.println("Enter height : ");
                height = sc.nextDouble();
        }

        public double calculateBMI(){
                return  weight/(height*height);

        }

        public String classifyBMI(double bmiValue){
```

```
        if(bmiValue <18.5) {
            return "Underweight";
        }
        else if(bmiValue <24.9) {
            return "Normal Weight";
        }
        else if(bmiValue<29.9) {
            return "overweight";
        }
        else {
            return "Obese";
        }
    }

    public void printRecord() {
        double bmi = calculateBMI();
        String classification = classifyBMI(bmi);

        System.out.println(bmi);
        System.out.println(classification);
    }

}


public class question3 {

    public static void main(String[] args) {
        bmi index = new bmi();

        index.acceptRecord();
        index.printRecord();
    }

}
```

4. Discount Calculation for Retail Sales

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

1. Accept the original price of an item and the discount percentage from the user.
2. 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`
3. Display the discount amount and the final price of the item, in Indian Rupees (₹).

ASSIGNMENT NO.3

Define class DiscountCalculator with methods acceptRecord, calculateDiscount & printRecord and test the functionality in main method.

Ans.

```
package org.example.ques4;
import java.util.Scanner;
class DiscountCalculator {
        int price;
        double discount;

        public void acceptRecord() {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the original price :");
                price = sc.nextInt();
                System.out.println("Enter the discount percentage :");
                discount = sc.nextDouble();
        }

        public void calculateDiscount() {
                double discountAmount = price*(discount/100);
                double finalPrice = price - discountAmount;
                printRecord(discountAmount, finalPrice);

        }
        public void printRecord(double discountAmount, double finalPrice) {
                System.out.println("Discount Amount: "+discountAmount);
                System.out.println("Final Amount: "+finalPrice);

        }
}

public class question4 {

        public static void main(String[] args) {
                DiscountCalculator  discAmt = new DiscountCalculator();

                discAmt.acceptRecord();
                discAmt.calculateDiscount();
//              discAmt.printRecord();

        }

}
```

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.

Ans.

```
package org.example.ques5;

import java.util.Scanner;

class TollBoothRevenueManager {
        double carTollRate;
        double truckTollRate;
        double motorcycleTollRate;
        int numberOfCars;
        int numberOfTrucks;
        int numberOfMotorcycles;
        public void acceptRecord() {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of cars :");
                numberOfCars = sc.nextInt();
                System.out.println("Enter the number of Trucks :");
                numberOfTrucks = sc.nextInt();
                System.out.println("Enter the number of Motorcycles :");
                numberOfMotorcycles = sc.nextInt();

        }

        public void setTollRates() {
                Scanner scanner = new Scanner(System.in);
     System.out.print("Enter the toll rate for cars (₹): ");
     carTollRate = scanner.nextDouble();
     System.out.print("Enter the toll rate for trucks (₹): ");
```

```
  truckTollRate = scanner.nextDouble();
  System.out.print("Enter the toll rate for motorcycles (₹): ");
  motorcycleTollRate = scanner.nextDouble();

    }

    public void calculateRevenue() {
        double totalRevenue = (numberOfCars*carTollRate)+(numberOfTrucks*truckTollRate)+(numberOfMotorcycles*motorcycleTollRate);
        printRecord(totalRevenue);
    }

    public void printRecord(double totalRevenue) {
        int totalVehicles = numberOfCars+numberOfMotorcycles+numberOfTrucks;
        System.out.println("the total number of vehicles :v" +totalVehicles);
        System.out.println("the total revenue : "+totalRevenue);

    }

}

public class question5 {

    public static void main(String[] args) {
        TollBoothRevenueManager toll = new TollBoothRevenueManager ();

        toll.carTollRate = 50.0;
        toll.truckTollRate = 100.0;
        toll.motorcycleTollRate = 30.0;


        toll.acceptRecord();
        toll.setTollRates();
        toll.calculateRevenue();
//      toll.printRecord();


    }

}
```