ASSIGNMENT NO.3

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.

```
package org.cdac;


import java.util.Scanner;

class LoanAmortizationCalculator {

   double principal, annualInterestRate, monthlyPayment, totalPayment;
   int loanTerm;


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

      System.out.print("Enter the loan amount (Principal): ₹");
      principal = sc.nextDouble();

      System.out.print("Enter the annual interest rate (in %): ");
      annualInterestRate = sc.nextDouble();

      System.out.print("Enter the loan term (in years): ");
      loanTerm = sc.nextInt();
   }


   public void calculateMonthlyPayment() {

      double monthlyInterestRate = annualInterestRate / 12 / 100;
```

```
        int numberOfMonths = loanTerm * 12;


        monthlyPayment = principal * (monthlyInterestRate * Math.pow(1 +
monthlyInterestRate, numberOfMonths))
                        / (Math.pow(1 + monthlyInterestRate, numberOfMonths) - 1);


        totalPayment = monthlyPayment * numberOfMonths;
    }
    public void printRecord() {
        System.out.println("\n Loan Amortization Report ");
        System.out.printf("Monthly Payment: ₹%.2f\n", monthlyPayment);
        System.out.printf("Total Amount Paid (Over the life of the loan): ₹%.2f\n",
totalPayment);
    }
}

class LoanAmortizationTest {
    public static void main(String[] args) {
        LoanAmortizationCalculator calc = new LoanAmortizationCalculator();
        calc.acceptRecord();
        calc.calculateMonthlyPayment();
        calc.printRecord();
    }
}
```

```
<terminated> LoanAmortizationTest [Java Application] C:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw
Enter the loan amount (Principal): ₹700
Enter the annual interest rate (in %): 4
Enter the loan term (in years): 2

 Loan Amortization Report
Monthly Payment: ₹30.40
Total Amount Paid (Over the life of the loan): ₹729.54
```

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`

ASSIGNMENT NO.3

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

```
package org.cdac;

import java.util.Scanner;

class CompoundInterestCalculator {
   double principal, annualInterestRate, futureValue, totalInterest;
   int numberOfCompounds, years;

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

      System.out.print("Enter the initial investment amount: ");
      principal = sc.nextDouble();

      System.out.print("Enter the annual interest rate (%): ");
      annualInterestRate = sc.nextDouble() / 100;

      System.out.print("Enter the number of times interest is compounded per year: ");
      numberOfCompounds = sc.nextInt();

      System.out.print("Enter the investment duration (years): ");
      years = sc.nextInt();
   }

   //  formula

   public void calculateFutureValue() {
      futureValue = principal * Math.pow((1 + annualInterestRate / numberOfCompounds),
numberOfCompounds * years);
      totalInterest = futureValue - principal;
   }

   // output
   public void printRecord() {
      System.out.printf("\nFuture Value: ₹%.2f\n", futureValue);
      System.out.printf("Total Interest Earned: ₹%.2f\n", totalInterest);
   }
}

class CompoundInterestTest {
```

```
    public static void main(String[] args) {
        CompoundInterestCalculator calc = new CompoundInterestCalculator();
        calc.acceptRecord();
        calc.calculateFutureValue();
        calc.printRecord();
    }
}
```

```
Enter the initial investment amount: 50000
Enter the annual interest rate (%): 8
Enter the number of times interest is compounded per year: 3
Enter the investment duration (years): 6

Future Value: ₹80297.18
Total Interest Earned: ₹30297.18
```

3. BMI (Body Mass Index) Tracker

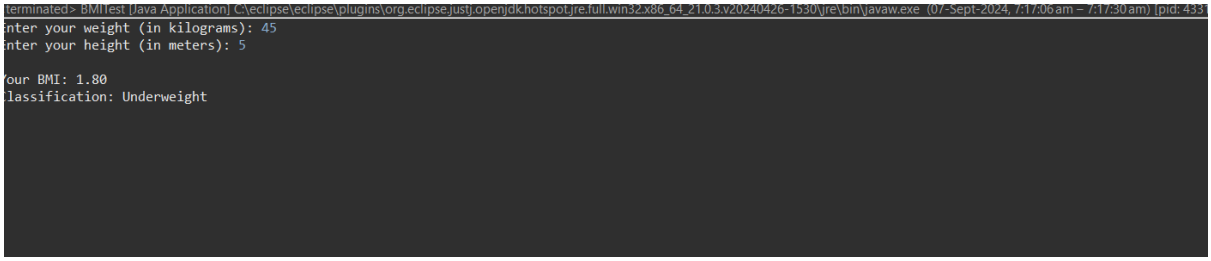
Create a system to calculate and classify Body Mass Index (BMI). The system should:

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.

```
package org.cdac.copy;
import java.util.Scanner;

class BMITracker {
    double weight, height, bmi;
    String classification;

    // Method to accept user inputs
    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter your weight (in kilograms): ");
        weight = sc.nextDouble();

        System.out.print("Enter your height (in meters): ");
        height = sc.nextDouble();
    }

    // Method to calculate BMI
    public void calculateBMI() {
        bmi = weight / (height * height);
    }

    // Method to classify BMI
    public void classifyBMI() {
        if (bmi < 18.5) {
            classification = "Underweight";
        } else if (bmi >= 18.5 && bmi < 24.9) {
            classification = "Normal Weight";
        } else if (bmi >= 25 && bmi < 29.9) {
            classification = "Overweight";
        } else {
            classification = "Obese";
        }
    }

    // Method to display the results
    public void printRecord() {
        System.out.printf("\nYour BMI: %.2f\n", bmi);
        System.out.println("Classification: " + classification);
    }
}

class BMITest {
    public static void main(String args[]) {
        BMITracker tracker = new BMITracker();
        tracker.acceptRecord();
        tracker.calculateBMI();
        tracker.classifyBMI();
        tracker.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 (₹).

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

```
package org.cdac;



import java.util.Scanner;

class DiscountCalculator {
   double originalPrice, discountRate, discountAmount, finalPrice;


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

      System.out.print("Enter the original price of the item (₹): ");
      originalPrice = sc.nextDouble();

      System.out.print("Enter the discount rate (%): ");
      discountRate = sc.nextDouble();
   }


   public void calculateDiscount() {
      discountAmount = originalPrice * (discountRate / 100);
```

```
        finalPrice = originalPrice - discountAmount;
    }


    public void printRecord() {
            System.out.println("Discount Amount: ₹" + discountAmount);
            System.out.println("Final Price: ₹" + finalPrice);

    }
}


class DiscountCalculatorTest {
    public static void main(String[] args) {
        DiscountCalculator calc = new DiscountCalculator();
        calc.acceptRecord();
        calc.calculateDiscount();
        calc.printRecord();
    }
}
```