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** Loan_Controler;

```
import java.util.Scanner;
class loan {
        private double principal;
        private double annual;
private int loanyear;
private  double monthlyPayment;
private  double totalPayment;
static Scanner sc=new Scanner(System.in);
public void accept() {
//  Scanner sc=new Scanner(System.in);
 System.out.println("Enter the Priciple");
 this.principal=sc.nextDouble();
 System.out.println("Enter the annual");
 this.annual=sc.nextDouble();
 System.out.println("Enter the loanyear");
 this.loanyear=sc.nextInt();
 sc.close();
```

```
}
public void Calculation() {

        int numberOfMonths = this.loanyear * 12;
        double monthlyInterestRate=this.annual/12/100;


  this. monthlyPayment = principal * (monthlyInterestRate * Math.pow(1 +
monthlyInterestRate, numberOfMonths)) /
                    (Math.pow(1 + monthlyInterestRate, numberOfMonths) - 1);
  this. totalPayment = this.monthlyPayment * numberOfMonths;
}
public void Print() {

        System.out.println("The Montly paymenr is " + this.monthlyPayment);
        System.out.println("The total payment is " + this.totalPayment);
}

}
public class question1 {
        public static void main(String[] args) {
                loan l=new loan();
                l.accept();
                l.Calculation();
                l.Print();
                loan.sc.close();

        }


}
```


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 (₹).

ASSIGNMENT NO.3

Define class CompoundInterestCalculator with methods acceptRecord , calculateFutureValue, printRecord and test the functionality in main method.

ANS:

```
package compound;

import java.util.Scanner;
public class question2 {
	private double investment ;
	private double annual;
private double compounded ;
private  double years;
private  double futureValue;
private  double totalInterest;
static Scanner sc=new Scanner(System.in);
public void Calculation() {
	 double rate = this.annual / 100;
   this. futureValue = this.investment * Math.pow((1 + rate / this.compounded),
this.compounded * this.years);
   this. totalInterest = futureValue - investment;
	}
public void accept() {
		//  Scanner sc=new Scanner(System.in);
		  System.out.println("Enter the investment");
		  this.investment=sc.nextDouble();
		  System.out.println("Enter the annual");
		  this.annual=sc.nextDouble();
		  System.out.println("Enter the compounded");
		  this.compounded=sc.nextDouble();
		  System.out.println("Enter the years");
		  this.years=sc.nextDouble();
		 }
public void Print() {

		System.out.println("The Montly paymenr is " + this.futureValue);
		System.out.println("The total payment is " + this.totalInterest);
	}
	public static void main(String[] args) {
		question2 l=new question2();
		l.accept();
		l.Calculation();
		l.Print();
		question2.sc.close();

	}
}
```

3. BMI (Body Mass Index) Tracker

ASSIGNMENT NO.3

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 BMI;
import java.util.Scanner;
public class Question3 {
        private double weight ;
        private double height;
private  double BMI ;
//private  double years;
//private  double futureValue;
//private  double totalInterest;
        static Scanner sc=new Scanner(System.in);


         public void Calculation() {
                this. BMI = weight / (height * height);
                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 < 29.9) {
                   System.out.println("Overweight");
               } else if (BMI >= 30) {
                   System.out.println("Obese");
               } else {
                   System.out.println("Invalid BMI value.");
               }
           }
         public void accept() {
                        //  Scanner sc=new Scanner(System.in);
                          System.out.println("Enter the weight in kg");
                          this.weight=sc.nextDouble();
```

```
                System.out.println("Enter the height in meter");
                this.height=sc.nextDouble();

            }

    public void Print() {

                System.out.println("The BMI paymenr is " + this.BMI);
    }



    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Question3 l=new Question3();
        l.accept();
        l.Calculation();
        l.Print();
        Question3.sc.close();
    }
}
```

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.

ANS:

```
package discount;
import java.util.Scanner;
public class Question4 {
        private double original ;
        private double discount;
private  double discountAmount ;
        private  double finalPrice ;
        static Scanner sc=new Scanner(System.in);
```

```
	public void accept() {
		//  Scanner sc=new Scanner(System.in);
		  System.out.println("Enter the original Price");
		  this.original=sc.nextDouble();
		  System.out.println("Enter the % in discount");
		  this.discount=sc.nextDouble();

		 }

	 public void Calculation() {
				this. discountAmount= this.original * (this.discount / 100);
				this.finalPrice = this.original - this.discountAmount;

	   }

	 public void Print() {

			System.out.println("The discountAmount  is " +  this.
discountAmount);
			System.out.println("The finalPrice  is " +  this. finalPrice);
	 }

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Question4 l=new Question4();
		l.accept();
		l.Calculation();
		l.Print();
		Question4.sc.close();
	}
}
```