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.

CODE:

```
package questions.org;
import java.util.Scanner;

//VarunVerma

class Loan{
	private float principal;
	private float interest;
	private float loan_term;
	private int m;
	private double monthlyPayment;
	private double amount_paid;

	public void takeInput() {
		Scanner scan = new Scanner(System.in);
		System.out.print("Enter the Loan Amount :      ");
		this.principal = scan.nextFloat();
		System.out.print("Enter the interest :  ");
		this.interest = scan.nextFloat();
		System.out.print("Enter the loan term(in Years) :    ");
		this.loan_term = scan.nextFloat();
		System.out.print("Enter the numbers of months :      ");
```

sandeepkulange@gmail.com

```
            this.m = scan.nextInt();

        }



        public void calculate() {

            // this.monthlyPayment = this.loan_amount * (this.interest *
Math.pow((1 + this.interest),this.loan_term)) / (Math.pow((1 +
this.interest),this.loan_term) - 1);
            float monthlyInterestRate = this.interest / 12 / 100;
            float numberOfMonths = this.loan_term * 12;
            this.monthlyPayment = this.principal * (monthlyInterestRate *
Math.pow((1 + monthlyInterestRate),numberOfMonths)) / (Math.pow((1 +
monthlyInterestRate),numberOfMonths) - 1);
            this.amount_paid = this.monthlyPayment * (this.loan_term * 12) +
this.m;
        }

        void displayOutput() {

            System.out.printf("The Monthly Payment is :  %.3f%n" ,
this.monthlyPayment);

            System.out.printf("Total amount paid over the life :%.3f%n",
this.amount_paid );

        }

}

public class loanCalculator {

        public static void main(String[] args) {
            // TODO Auto-generated method stub
            Loan payment = new Loan();
            payment.takeInput();
            payment.calculate();
            payment.displayOutput();
        }

}
```

sandeepkulange@gmail.com

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.

CODE:

```
package compound.org;
import java.util.Scanner;



class Invest{
	float principal_amount;
	float annualInterestRate;
	int compoundPerYear;
	int investmentDuration;
	double futureValue;
	double totalInterest;


	void acceptRecord() {
		Scanner scan = new Scanner(System.in);
		System.out.print("Enter the Initial Investment Amount : ");
		this.principal_amount = scan.nextFloat();
		System.out.print("Enter the Annual Interest Rate : ");
		this.annualInterestRate = scan.nextFloat();
		System.out.print("Enter the numeber of times interest is 
compounded : ");
		this.compoundPerYear = scan.nextInt();
		System.out.print("Enter the Investment Duration : ");
		this.investmentDuration = scan.nextInt();


	}


	void futureValueCalculator() {

		this.futureValue = this.principal_amount * (1 + 
this.annualInterestRate / Math.pow((this.compoundPerYear), 
this.investmentDuration));
		  this.totalInterest = this.futureValue - this.principal_amount;
	}



	void printRecord() {

		System.out.println("The Future Value is : " + this.futureValue);
		System.out.println("Total Interest earned is : " + 
this.totalInterest);


	}
```

sandeepkulange@gmail.com

```
}

public class CompoundInterest {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                Invest inv = new Invest();
                inv.acceptRecord();
                inv.futureValueCalculator();
                inv.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.

CODE:

```
package org.bmi;
import java.util.Scanner;

class Tracker{
        float weight;
        float height;
        float bmassIndex;

        public void acceptRecord() {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter weight");
                this.weight = scan.nextFloat();
                System.out.println("Enter Height");
                this.height = scan.nextFloat();

        }

        public void calculate() {
```

sandeepkulange@gmail.com

```
        this.bmassIndex = this.weight / (this.height * this.height);


    }

    public void printRecord() {
        if (this.bmassIndex < 18.5) {
            System.out.println("Under Weight");
        }
        else if (this.bmassIndex >= 18.5 && this.bmassIndex <= 24.9) {
            System.out.println("Normal Weight");
        }
        else if (this.bmassIndex >= 25  && this.bmassIndex <= 29.9) {
            System.out.println("Over Weight");
        }
        else {
            System.out.println("Obese");
        }
    }
}


public class bmi {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Tracker track = new Tracker();
        track.acceptRecord();
        track.calculate();
        track.printRecord();
    }

}
```

sandeepkulange@gmail.com

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.

CODE:

```java
package org.toll;
import java.util.Scanner;

class DiscountCalculator{
        float price;
        float discount;
        float discountAmount;
        float finalPrice;

        public void acceptRecord() {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the price");
                this.price = scan.nextFloat();
                System.out.println("Enter the discount");
                this.discount = scan.nextFloat();
        }

        public void calculateDiscount() {
                this.discountAmount = this.price * (this.discount / 100);
                this.finalPrice = this.price - this.discountAmount;
        }

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

        }

}

public class Toll {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                DiscountCalculator disc = new DiscountCalculator();

                disc.acceptRecord();
                disc.calculateDiscount();
                disc.printRecord();
        }

}
```

sandeepkulange@gmail.com

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.

CODE:

```
package org.rate;
import java.util.Scanner;

class TollBoothRevenueManager{
        int vehicleNumber;  //non-static field
        float car;
        float truck;
        float motorcycle;
        double totalRevenue;
        int motorcycleNumber;
        int carNumber;
        int truckNumber;

        public void acceptRecord() {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the number of bike");
                this.motorcycleNumber = scan.nextInt();
                System.out.println("Enter the number of car");
                this.carNumber = scan.nextInt();
                System.out.println("Enter the number of truck");
                this.truckNumber = scan.nextInt();
                //scan.close();
        }

        public void setTollRate() {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the Toll Rate of CAR");
                this.car = scan.nextFloat();

                System.out.println("Enter the Toll Rate of TRUCK");
                this.truck = scan.nextFloat();

                System.out.println("Enter the Toll Rate of BIKE");
                this.motorcycle = scan.nextFloat();
                //scan.close();
        }

        public void calculateRevenue() {

                this.totalRevenue = ((this.car * this.carNumber) + (this.truck *
this.truckNumber) + (this.motorcycle * this.motorcycleNumber));

                this.vehicleNumber = this.carNumber + this.motorcycleNumber +
this.truckNumber;
```

sandeepkulange@gmail.com

```
        }

        public void printRecord() {

                System.out.println("Total Revenue is " + this.totalRevenue);
                System.out.println("Number of Vehicles is " + this.vehicleNumber);

        }

}

public class TollPlaza {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                TollBoothRevenueManager manage = new TollBoothRevenueManager();
//instance
                manage.setTollRate();
                manage.acceptRecord();
                manage.calculateRevenue();
                manage.printRecord();
        }

}
```

sandeepkulange@gmail.com