

Sandeep Sir's : Assignment 4

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:

o Monthly Payment Calculation:

$$\text{monthlyPayment} = \text{principal} * (\text{monthlyInterestRate} * (1 + \text{monthlyInterestRate})^{\text{numberOfMonths}}) / ((1 + \text{monthlyInterestRate})^{\text{numberOfMonths}} - 1)$$

Where $\text{monthlyInterestRate} = \text{annualInterestRate} / 12 / 100$ and $\text{numberOfMonths} = \text{loanTerm} * 12$

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 the class `LoanAmortizationCalculator` with fields, an appropriate constructor, getter and setter methods, a `toString` method and business logic methods. Define the class `LoanAmortizationCalculatorUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method and test the functionality of the utility class.

→ `LoanAmortizationCalculator`:

```
package com.assignment4.question1;

public class LoanAmortizationCalculator {
    private double principalAmount;
    private double interestRate;
    private double loanTerm;
    private double monthlyPayment;
    private double totalAmountPaid;
```

```

private double monthlyInterestRate;
private double numberOfMonths;

public LoanAmortizationCalculator() {
}

LoanAmortizationCalculator(double p, double i , double
1) {
    // TODO Auto-generated constructor stub
    this.principalAmount = p;
    this.interestRate = i;
    this.loanTerm = 1;
}

public void calculateMonthlyPayment() {
    this.numberOfMonths = this.loanTerm * 12;
    this.monthlyInterestRate = this.interestRate / 12 /
100;
    this.monthlyPayment = this.principalAmount
        * (this.monthlyInterestRate * Math.pow(1 +
this.monthlyInterestRate, this.numberOfMonths))
        / (Math.pow(1 + this.monthlyInteres
tRate, this.numberOfMonths) - 1);
    this.totalAmountPaid = this.numberOfMonths * this.m
onthlyPayment;
}

public double getPrincipalAmount() {
    return principalAmount;
}

public void setPrincipalAmount(double principalAmount)
{
    this.principalAmount = principalAmount;
}

```

```

    public double getInterestRate() {
        return interestRate;
    }

    public void setInterestRate(double interestRate) {
        this.interestRate = interestRate;
    }

    public double getLoanTerm() {
        return loanTerm;
    }

    public void setLoanTerm(double loanTerm) {
        this.loanTerm = loanTerm;
    }

    @Override
    public String toString() {
        this.calculateMonthlyPayment();
        return "LoanAmortizationCalculator [totalAmountPaid =" + totalAmountPaid + ", monthlyPayment=" + monthlyPayment + "]";
    }
}

```

LoanAmortizationCalculatorUtil:

```

package com.assignment4.question1;

import java.util.Scanner;

public class LoanAmortizationCalculatorUtil {

    private LoanAmortizationCalculator l = new LoanAmortizationCalculator();
}

```

```

    public static int menuList(Scanner sc) {
        System.out.println("0.Exit.");
        System.out.println("1.Accept Record.");
        System.out.println("2.Print Record.");
        System.out.print("Enter choice : ");
        int choice = sc.nextInt( );
        return choice;
    }

    public void acceptRecord(Scanner sc) {
        System.out.println("Enter Principal Amount: ");
        l.setPrincipalAmount(sc.nextDouble());
        System.out.println("Enter Interest Rate: ");
        l.setInterestRate(sc.nextDouble());
        System.out.println("Enter Loan Term in years: ");
        l.setLoanTerm(sc.nextDouble());
    }

    public void printRecord() {
        System.out.println(l.toString());
    }
}

```

Program.java:

```

package com.assignment4.question1;

import java.util.Scanner;

public class question1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        LoanAmortizationCalculatorUtil u1 = new LoanAmortiz
ationCalculatorUtil();
    }
}

```

```

        int choice;;
        while ( ( choice = u1.menuList( sc ) ) != 0 ) {
            switch( choice ) {
                case 1:
                    u1.acceptRecord(sc);
                    break;
                case 2:
                    u1.printRecord();
                    break;
            }
        }
    }
}

```

```

0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      1
Enter Principal Amount:
10000
Enter Interest Rate:
24
Enter Loan Term in years:
1
0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      2
LoanAmortizationCalculator [totalAmountPaid=11347.151594754172, monthlyPayment=945.5959662295144]

```

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:

oFuture Value Calculation:

$$\text{futureValue} = \text{principal} * (1 + \text{annualInterestRate} / \text{numberOfCompounds})^{(\text{numberOfCompounds} * \text{years})}$$
 oTotal Interest Earned: $\text{totalInterest} = \text{futureValue} - \text{principal}$
 3.Display the future value and the total interest earned, in Indian Rupees (₹).
 Define the class CompoundInterestCalculator with fields, an appropriate constructor, getter and setter methods, a toString method and business logic methods. Define the class CompoundInterestCalculatorUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

→ CompoundInterestCalculator:

```

package com.assignment4.question2;

public class CompoundInterestCalculator {
    private double is;
    private double ir;
    private int n;
    private int d;
    private double futureValue;
    private double totalInterest;

    CompoundInterestCalculator(){};

    public void calculateFutureValue() {
        this.futureValue = this.is * Math.pow((1 + this.ir / this.n), (this.n * this.d));
        this.totalInterest = this.futureValue - this.is;
    }

    @Override
    public String toString() {
        calculateFutureValue();
        return "CompoundInterestCalculator [futureValue=" + futureValue + ", totalInterest=" + totalInterest + "];"
    }
  
```

```
    public double getIs() {  
        return is;  
    }  
  
    public void setIs(double is) {  
        this.is = is;  
    }  
  
    public double getIr() {  
        return ir;  
    }  
  
    public void setIr(double ir) {  
        this.ir = ir;  
    }  
  
    public int getN() {  
        return n;  
    }  
  
    public void setN(int n) {  
        this.n = n;  
    }  
  
    public int getD() {  
        return d;  
    }  
  
    public void setD(int d) {  
        this.d = d;  
    }  
}
```

CompoundInterestCalculatorUtil

```

package com.assignment4.question2;

import java.util.Scanner;

public class CompoundInterestCalculatorUtil {
    CompoundInterestCalculator c = new CompoundInterestCalculator();

    public static int menuList(Scanner sc) {
        System.out.println("0.Exit.");
        System.out.println("1.Accept Record.");
        System.out.println("2.Print Record.");
        System.out.print("Enter choice : ");
        int choice = sc.nextInt();
        return choice;
    }

    public void acceptRecord(Scanner sc) {
        System.out.println("ENTER INITIAL INVESTMENT AMOUNT: ");
        c.setIs(sc.nextDouble());
        System.out.println("ENTER Annual Interest Rate: ");
        c.setIr(sc.nextDouble());
        System.out.println("ENTER INTEREST COMPOUND TIME: ");
        c.setN(sc.nextInt());
        System.out.println("ENTER INVESTMENT DURATION in years: ");
        c.setD(sc.nextInt());
    }

    public void printRecord() {
        System.out.println(c.toString());
    }
}

```



```
}  
}
```

Program.java

```
package com.assignment4.question2;  
  
import java.util.Scanner;  
  
public class Question2 {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Scanner sc = new Scanner(System.in);  
        CompoundInterestCalculatorUtil c1 = new CompoundInterestCalculatorUtil();  
  
        int choice;  
        while ( ( choice = c1.menuList( sc ) ) != 0 ) {  
            switch( choice ) {  
                case 1:  
                    c1.acceptRecord(sc);  
                    break;  
                case 2:  
                    c1.printRecord();  
                    break;  
            }  
        }  
    }  
}
```

```

0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      1
ENTER INITIAL INVESTMENT AMOUNT:
10000
ENTER Annual Interest Rate:
24
ENTER INTEREST COMPOUND TIME:
3
ENTER INVESTMENT DURATION in years:
1
0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      2
CompoundInterestCalculator [futureValue=7290000.0, totalInterest=7280000.0]

```

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:

oBMI Calculation: $BMI = \text{weight} / (\text{height} * \text{height})$

3.Classify the BMI into one of the following categories:

oUnderweight: $BMI < 18.5$

oNormal weight: $18.5 \leq BMI < 24.9$

oOverweight: $25 \leq BMI < 29.9$

oObese: $BMI \geq 30$

4.Display the BMI value and its classification.

Define the class BMITracker with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic methods. Define the class BMITrackerUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

→ BMITracker:

```

package com.assignment4.question3;

public class BMITracker {
    private double height;
    private double weight;
    private double BMI;

    BMITracker() {
        // TODO Auto-generated constructor stub
    }

    public double getHeight() {
        return height;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    void calculateBMI() {
        this.BMI = this.weight / (this.height * this.height);
    }

    String classifyBMI() {
        calculateBMI();
        if(this.BMI>=30) {

```

```

        return "Obese";
    }else if(this.BMI >= 25) {
        return "Overweight";
    }else if(this.BMI >= 18.5) {
        return "Normal Weight";
    }else if(this.BMI>0) {
        return "Underweight";
    }else {
        return "Enter Valid Input";
    }
}

@Override
public String toString() {
    return classifyBMI();
}
}

```

BMITrackerUtil:

```

package com.assignment4.question3;

import java.util.Scanner;

public class BMITrackerUtil {
    private BMITracker b = new BMITracker();

    public static int menuList(Scanner sc) {
        System.out.println("\n\n\n");
        System.out.println("0.Exit.");
        System.out.println("1.Accept Record.");
        System.out.println("2.Print Record.");
        System.out.print("Enter choice : ");
        int choice = sc.nextInt( );
        return choice;
    }
}

```

```

        public void acceptRecord(Scanner sc) {
            System.out.println("ENTER Height: ");
            b.setHeight(sc.nextDouble());
            System.out.println("ENTER Weight: ");
            b.setWeight(sc.nextDouble());
        }

        public void printRecord() {
            System.out.println(b.toString());
        }
    }
}

```

Program.java

```

package com.assignment4.question3;

import java.util.Scanner;

public class Question3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        BMITrackerUtil b1 = new BMITrackerUtil();

        int choice;
        while ( ( choice = b1.menuList( sc ) ) != 0 ) {
            switch( choice ) {
                case 1:
                    b1.acceptRecord(sc);
                    break;
                case 2:
                    b1.printRecord();
                    break;
            }
        }
    }
}

```

```
}  
}  
}
```

```
0.Exit.  
1.Accept Record.  
2.Print Record.  
Enter choice      :      1  
ENTER Height:  
1.77  
ENTER Weight:  
96  
  
0.Exit.  
1.Accept Record.  
2.Print Record.  
Enter choice      :      2  
Obese
```

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:

oDiscount Amount Calculation: $\text{discountAmount} = \text{originalPrice} * (\text{discountRate} / 100)$

oFinal Price Calculation: $\text{finalPrice} = \text{originalPrice} - \text{discountAmount}$

3.Display the discount amount and the final price of the item, in Indian Rupees (₹).

Define the class DiscountCalculator with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic methods. Define the class DiscountCalculatorUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

→ DiscountCalculator

```
package com.assignment4.question4;

public class DiscountCalculator {
    private double ogPrice;
    private double discountRate;
    private double discountAmount;
    private double finalPrice;

    public DiscountCalculator() {
        // TODO Auto-generated constructor stub
    }

    public double getOgPrice() {
        return ogPrice;
    }

    public void setOgPrice(double ogPrice) {
        this.ogPrice = ogPrice;
    }

    public double getDiscountRate() {
        return discountRate;
    }

    public void setDiscountRate(double discountRate) {
        this.discountRate = discountRate;
    }
}
```

```

        void calculateDiscount () {
            this.discountAmount = this.ogPrice * (this.discount
Rate / 100);
            this.finalPrice = this.ogPrice - this.discountAmoun
t;
        }

        @Override
        public String toString() {
            calculateDiscount ();
            return "DiscountCalculator [discountAmount=" + disc
ountAmount + ", finalPrice=" + finalPrice + "]";
        }
    }
}

```

DiscountCalculatorUtil

```

package com.assignment4.question4;

import java.util.Scanner;

public class DiscountCalculatorUtil {
    private DiscountCalculator d = new DiscountCalculator
();

    public static int menuList(Scanner sc) {
        System.out.println("\n");
        System.out.println("0.Exit.");
        System.out.println("1.Accept Record.");
        System.out.println("2.Print Record.");
        System.out.print("Enter choice : ");
        int choice = sc.nextInt( );
        return choice;
    }
}

```



```

        public void acceptRecord(Scanner sc) {
            System.out.println("ENTER Original Price: ");
            d.setOgPrice(sc.nextDouble());
            System.out.println("ENTER Discount Rate: ");
            d.setDiscountRate(sc.nextDouble());
        }

        public void printRecord() {
            System.out.println(d.toString());
        }
    }
}

```

Program.java

```

package com.assignment4.question4;

import java.util.Scanner;

public class Question4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        DiscountCalculatorUtil d1 = new DiscountCalculatorUtil();

        int choice;
        while ( ( choice = d1.menuList( sc ) ) != 0 ) {
            switch( choice ) {
                case 1:
                    d1.acceptRecord(sc);
                    break;
                case 2:
                    d1.printRecord();
                    break;
            }
        }
    }
}

```

```

    }
}
}

```

```

0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      1
ENTER Original Price:
1000
ENTER Discount Rate:
10

0.Exit.
1.Accept Record.
2.Print Record.
Enter choice      :      2
DiscountCalculator [discountAmount=100.0, finalPrice=900.0]

```

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:

oCar: ₹50.00

oTruck: ₹100.00

oMotorcycle: ₹30.00

Define the class TollBoothRevenueManager with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic

methods. Define the class TollBoothRevenueManagerUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

→ TollBoothRevenueManager:

```
package com.assignment4.question5;

import java.util.Scanner;

public class TollBoothRevenueManager {
    private double carRate;
    private int carNumber;
    private double truckRate;
    private int truckNumber;
    private double motorcycleRate;
    private int motorcycleNumber;
    private double carRevenue;
    private double truckRevenue;
    private double motorcycleRevenue;

    public TollBoothRevenueManager() {
        // TODO Auto-generated constructor stub
    }

    void calculateRevenue() {
        this.carRevenue = this.carRate * this.carNumber;
        this.truckRevenue = this.truckRate * this.truckNumber;
        this.motorcycleRevenue = this.motorcycleRate * this.motorcycleNumber;
    }

    public double getCarRate() {
        return carRate;
    }
}
```

```
}

public void setCarRate(double carRate) {
    this.carRate = carRate;
}

public int getCarNumber() {
    return carNumber;
}

public void setCarNumber(int carNumber) {
    this.carNumber = carNumber;
}

public double getTruckRate() {
    return truckRate;
}

public void setTruckRate(double truckRate) {
    this.truckRate = truckRate;
}

public int getTruckNumber() {
    return truckNumber;
}

public void setTruckNumber(int truckNumber) {
    this.truckNumber = truckNumber;
}

public double getMotorcycleRate() {
    return motorcycleRate;
}

public void setMotorcycleRate(double motorcycleRate) {
    this.motorcycleRate = motorcycleRate;
}
```

```

    }

    public int getMotorcycleNumber() {
        return motorcycleNumber;
    }

    public void setMotorcycleNumber(int motorcycleNumber) {
        this.motorcycleNumber = motorcycleNumber;
    }

    @Override
    public String toString() {
        calculateRevenue();
        return "TollBoothRevenueManager [carRevenue=" + car
Revenue + ", truckRevenue=" + truckRevenue
        + ", motorcycleRevenue=" + motorcycleRevenue
e + "]\n";
    }
}

```

TollBoothRevenueManagerUtil

```

package com.assignment4.question5;

import java.util.Scanner;

public class TollBoothRevenueManagerUtil {
    private TollBoothRevenueManager d = new TollBoothRevenueManager();

    public static int menuList(Scanner sc) {
        System.out.println("\n");
        System.out.println("0.Exit.");
        System.out.println("1.Enter Rates.");
        System.out.println("2.Accept Record.");
        System.out.println("3.Print Record.");
    }
}

```

```

        System.out.print("Enter choice : ");
        int choice = sc.nextInt( );
        return choice;
    }

    void setTollRates(Scanner sc){
        System.out.println("Enter Rate per Car: ");
        d.setCarRate(sc.nextDouble());
        System.out.println("Enter Rate per Truck: ");
        d.setTruckRate(sc.nextDouble());
        System.out.println("Enter Rate per Motorcycle: ");
        d.setMotorcycleRate(sc.nextDouble());
    }

    public void acceptRecord(Scanner sc) {
        System.out.println("Enter Number of Cars: ");
        d.setCarNumber(sc.nextInt());
        System.out.println("Enter Number of Truck: ");
        d.setTruckNumber(sc.nextInt());
        System.out.println("Enter Number of Motorcycle: ");
        d.setMotorcycleNumber(sc.nextInt());
    }

    public void printRecord() {
        System.out.println(d.toString());
    }
}

```

Program.java

```

package com.assignment4.question5;

import java.util.Scanner;

public class Question5 {

```

```

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    TollBoothRevenueManagerUtil d1 = new TollBoothRevenueManagerUtil();

    int choice;
    while ( ( choice = d1.menuList( sc ) ) != 0 ) {
        switch( choice ) {
            case 1:
                d1.setTollRates(sc);
                break;
            case 2:
                d1.acceptRecord(sc);
                break;
            case 3:
                d1.printRecord();
                break;
        }
    }
}

```

```
0.Exit.
1.Enter Rates.
2.Accept Record.
3.Print Record.
Enter choice      :      1
Enter Rate per Car:
10
Enter Rate per Truck:
20
Enter Rate per Motorcycle:
5

0.Exit.
1.Enter Rates.
2.Accept Record.
3.Print Record.
Enter choice      :      2
Enter Number of Cars:
10
Enter Number of Truck:
5
Enter Number of Motorcycle:
20

0.Exit.
1.Enter Rates.
2.Accept Record.
3.Print Record.
Enter choice      :      3
TollBoothRevenueManager [carRevenue=100.0, truckRevenue=100.0, motorcycleRevenue=100.0]
```
