

Sandeep Sir's : Assignment 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:

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 class `LoanAmortizationCalculator` with methods `acceptRecord`, `calculateMonthlyPayment` & `printRecord` and test the functionality in main method.

→

```
package com.assignment3.question1;

import java.util.Scanner;

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

```

private double monthlyInterestRate;
private double numberOfMonths;

public void acceptRecord(Scanner sc) {
    System.out.println("Enter Principal Amount: ");
    this.principalAmount = sc.nextLong();
    System.out.println("Enter Interest Rate: ");
    this.interestRate = sc.nextDouble();
    System.out.println("Enter Loan Term in years: ");
    this.loanTerm = sc.nextInt();
}

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 void printRecord() {
    System.out.println("Monthly payment is: " + this.mo
nthlyPayment);
    System.out.println("Total Amount paid is: " + this.
totalAmountPaid);
}

}

public class Question1 {

```

```

    public static void main(String[] args) {
        // Loan Amortization Calculator
        Scanner sc = new Scanner(System.in);
        LoanAmortizationCalculator l1 = new LoanAmortizationCalculator();
        l1.acceptRecord(sc);
        l1.calculateMonthlyPayment();
        l1.printRecord();
    }
}

```

```

Enter Principal Amount:
10000
Enter Interest Rate:
24
Enter Loan Term in years:
1
Monthly payment is: 945.5959662295144
Total Amount paid is: 11347.151594754172

```

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

→

```
package com.assignment3.question2;

import java.util.Scanner;

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

    public void acceptRecord(Scanner sc) {
        System.out.println("ENTER INITIAL INVESTMENT AMOUNT: ");
        this.is = sc.nextDouble();
        System.out.println("ENTER annual interest rate: ");
        this.ir = sc.nextDouble();
        System.out.println("ENTER INTEREST COMPOUND TIME: ");
        this.n = sc.nextInt();
        System.out.println("ENTER INVESTMENT DURATION in years: ");
        this.d = sc.nextInt();
    }

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

```

    }

    public void returnRecord() {
        this.calculateFutureValue();
        System.out.println("Future value is: " + this.futureValue);
        System.out.println("Total Interest is: " + this.totalInterest);
    }
}

public class Qurstion2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        CompoundInterestCalculator e1 = new CompoundInterestCalculator();

        e1.acceptRecord(sc);
        e1.returnRecord();

        sc.close();

    }
}

```

```
ENTER INITIAL INVESTMENT AMOUNT:
10000
ENTER annual interest rate:
24
ENTER INTEREST COMPOUND TIME:
3
ENTER INVESTMENT DURATION in years:
1
Future value is: 7290000.0
Total Interest is: 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:

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

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

o Underweight: $BMI < 18.5$

o Normal weight: $18.5 \leq BMI < 24.9$

o Overweight: $25 \leq BMI < 29.9$

o Obese: $BMI \geq 30$

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 com.assignment3.question3;

import java.util.Scanner;

class BMITracker{
    private double height;
    private double weight;
```

```

private double BMI;

void acceptRecord(Scanner sc) {
    System.out.println("Enter Height: ");
    this.height = sc.nextDouble();
    System.out.println("Enter Weight: ");
    this.weight = sc.nextDouble();
}

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";
    }
}

void printRecord() {
    String str = this.classifyBMI();
    System.out.println(str);
}
}

```

```

public class Question3 {

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

        b1.acceptRecord(sc);
        b1.printRecord();

    }

}

```

```

<terminated> Question3 [Java Applic
Enter Height:

1.77
Enter Weight:
96
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 class DiscountCalculator with methods acceptRecord, calculateDiscount & printRecord and test the functionality in main method.

→

```
package com.assignment3.question4;

import java.util.Scanner;

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

    void acceptRecord(Scanner sc) {
        System.out.println("Enter Original Price: ");
        this.ogPrice = sc.nextDouble();
        System.out.println("Enter Discount rate: ");
        this.discountRate = sc.nextDouble();
    }

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

    void printRecord() {
        calculateDiscount();
        System.out.println("Discount amount: " + this.disco
untAmount);
        System.out.println("Total Amount paid is: " + this.
```

```

finalPrice);
    }
}

public class Question4 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DiscountCalculator a1 = new DiscountCalculator();
        a1.acceptRecord(sc);
        a1.printRecord();
    }

}

```

```

<terminated> Question4 [Java Application] C:\User
Enter Original Price:
1000
Enter Discount rate:
10
Discount amount: 100.0
Total Amount paid is: 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 class TollBoothRevenueManager with methods acceptRecord, setTollRates, calculateRevenue & printRecord and test the functionality in main method.

→

```
package com.assignment3.question5;

import java.util.Scanner;

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;

    void acceptRecord(Scanner sc) {
        System.out.println("Enter Number of Cars: ");
        this.carNumber = sc.nextInt();
        System.out.println("Enter Number of Truck: ");
        this.truckNumber= sc.nextInt();
        System.out.println("Enter Number of Motorcycle: ");
        this.motorcycleNumber= sc.nextInt();
    }

    void setTollRates(Scanner sc){
```

```

        System.out.println("Enter Rate per Car: ");
        this.carRate = sc.nextDouble();
        System.out.println("Enter Rate per Truck: ");
        this.truckRate= sc.nextDouble();
        System.out.println("Enter Rate per Motorcycle: ");
        this.motorcycleRate= sc.nextDouble();
    }

    void calculateRevenue() {
        this.carRevenue = this.carRate * this.carNumber;
        this.truckRevenue = this.truckRate * this.truckNumb
er;
        this.motorcycleRevenue = this.motorcycleRate * thi
s.motorcycleNumber;
    }

    void printRecord() {
        calculateRevenue();
        System.out.println("Total Revenue is: ");
        System.out.println("Car: " + this.carRevenue);
        System.out.println("Truck: " + this.truckRevenue);
        System.out.println("Motorcycle: " + this.motorcycle
Revenue);
    }
}

public class Question5 {

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

        TollBoothRevenueManager t1 = new TollBoothRevenueMa
nager();

        t1.setTollRates(sc);
    }
}

```

```
        t1.acceptRecord(sc);
        t1.printRecord();

    }

}
```

terminated: questions para Application 3

```
Enter Rate per Car:
5
Enter Rate per Truck:
30
Enter Rate per Motorcycle:
2
Enter Number of Cars:
10
Enter Number of Truck:
10
Enter Number of Motorcycle:
10
Total Revenue is:
Car: 50.0
Truck: 300.0
Motorcycle: 20.0
```
