**Assignment-3**

**079\_Raj kumar\_KH**

**1. Loan Amortization Calculator**

Implement a system to calculate and display the monthly payments for a mortgage loan. The system should:

* Accept the principal amount (loan amount), annual interest rate, and loan term (in years) from the user.
* 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
* 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** assignment3.org;

**import** java.util.Scanner;

**class** LoanAmortizationCalculator{

**double** principal;

**double** annualInterestRate;

**int** loanTerm;

**double** monthlyPayment;

**double** totalAmountPaid;

Scanner sc = **new** Scanner(System.***in***);

**public** **void** acceptRecord() {

System.***out***.println("Enter Principle Amount:₹ ");

**this**.principal = sc.nextDouble();

System.***out***.println("Enter annual interest rate: ");

**this**.annualInterestRate = sc.nextDouble();

System.***out***.println("Enter loan term: ");**this**.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);

totalAmountPaid = monthlyPayment \* loanTerm \* 12;

}

**public** **void** printRecord() {

System.***out***.println("Monthly Payment: ₹" + monthlyPayment);

System.***out***.println("Total Payment: ₹ "+totalAmountPaid );

}

}

**public** **class** Assignment3 {

**public** **static** **void** main(String[] args) {

LoanAmortizationCalculator loan = **new** LoanAmortizationCalculator ();

loan.acceptRecord();

loan.calculateMonthlyPayment();

loan.printRecord();

}

}



**2. Compound Interest Calculator for Investment**

Develop a system to compute the future value of an investment with compound interest. The system should:

* 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.
* 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
* 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** assignment3.org;

**import** java.util.Scanner;

**class** CompoundInterestCalculator {

**int** principle;

**double** annualInterestRate;

**int** numberOfCompounds;

**int** years;

**double** futureValue;

**double** totalInterest;

**public** **void** acceptRecord() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Principle :");

**this**.principle = sc.nextInt();

System.***out***.println("Annual Interest Rate :");

**this**.annualInterestRate = sc.nextDouble() / 100;

System.***out***.println("Number of Compounds :");

**this**.numberOfCompounds = sc.nextInt();

System.***out***.println("Years :");

**this**.years = sc.nextInt();

sc.close();

}

**public** **void** calculateFutureValue() {

futureValue = principle \* Math.*pow*(1 + annualInterestRate /

numberOfCompounds, numberOfCompounds \* years);

totalInterest = futureValue - principle;}

**public** **void** printRecord() {

System.***out***.println("Future Value : "+futureValue);

System.***out***.println("Total Interest Earned : "+totalInterest);

}

}

**public** **class** project {

**public** **static** **void** main(String[] args) {

CompoundInterestCalculator calculator = **new** CompoundInterestCalculator();

calculator.acceptRecord();

calculator.calculateFutureValue();

calculator.printRecord();

}

}



**3. BMI (Body Mass Index) Tracker**

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

* Accept weight (in kilograms) and height (in meters) from the user.
* Calculate the BMI using the formula:
* **BMI Calculation:** BMI = weight / (height \* height)
* 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
* 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.example;

**import** java.util.Scanner;

**public** **class** BMI {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter your height in meters: ");

**double** height = scanner.nextDouble();

System.***out***.print("Enter your weight in kilograms: ");

**double** weight = scanner.nextDouble();

**double** bmi = weight / (height \* height);

String category;

**if** (bmi < 18.5) {

category = "Underweight";

} **else** **if** (bmi >= 18.5 && bmi < 24.9) {

category = "Normal weight";

} **else** **if** (bmi >= 25 && bmi < 29.9) {

category = "Overweight";

} **else** {

category = "Obesity";

}

System.***out***.printf("Your BMI is %.2f, which is classified as %s.%n", bmi, category);

scanner.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:

* Accept the original price of an item and the discount percentage from the user.
* 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
* 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** assignment3.org;

**import** java.util.Scanner;

**class** DiscountCalculator {

**float** originalPrice;

**float** discountRate;

**float** finalPrice;

**float** discountAmount;

**public** **void** acceptRecord() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter Original Price: ");

**this**.originalPrice = sc.nextFloat();

System.***out***.println("Enter Discount Rate: ");

**this**.discountRate = sc.nextFloat();

sc.close();

}

**public** **void** calculateDiscount() {

discountAmount = originalPrice \* (discountRate / 100);

finalPrice = originalPrice - discountAmount;

}

**public** **void** printRecord() {

System.***out***.println("Discount Amount: "+ discountAmount);

System.***out***.println("Final Amount: "+ finalPrice);

}

}

**public** **class** project2 {

**public** **static** **void** main(String[] args) {

DiscountCalculator cal = **new** DiscountCalculator();

cal.acceptRecord();

cal.calculateDiscount();

cal.printRecord();

}

}



**5. Toll Booth Revenue Management**

Develop a system to simulate a toll booth for collecting revenue. The system should:

* Allow the user to set toll rates for different vehicle types: Car, Truck, and Motorcycle.
* Accept the number of vehicles of each type passing through the toll booth.
* Calculate the total revenue based on the toll rates and number of vehicles.
* 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.

**package** assignment3.org;

**import** java.util.Scanner;

**class** TollBoothRevenueManager{

**private** **double** carTollRate;

**private** **double** truckTollRate;

**private** **double** motorcycleTollRate;

**private** **int** carCount;

**private** **int** truckCount;

**private** **int** motorcycleCount;

**private** **double** totalRevenue;

**public** **void** setTollRates() {

Scanner sc = **new** Scanner(System.***in***);System.***out***.println("Enter toll rate for Car:");

carTollRate = sc.nextDouble();

System.***out***.println("Enter the toll rate for Truck:");

truckTollRate = sc.nextDouble();

System.***out***.println("Enter the toll rate for Motorcycle:");

motorcycleTollRate = sc.nextDouble();

}

**public** **void** acceptRecord() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the number of Cars:");

carCount = sc.nextInt();

System.***out***.println("Enter the number of Trucks:");

truckCount = sc.nextInt();

System.***out***.println("Enter the number of Motorcycles:");

motorcycleCount = sc.nextInt();

}

**public** **void** calculateRevenue() {

totalRevenue = (carCount \* carTollRate) + (truckCount \* truckTollRate) +

(motorcycleCount \* motorcycleTollRate);

}

**public** **void** printRecord() {

**int** totalVehicles = carCount + truckCount + motorcycleCount;

System.***out***.println("Total number of vehicles: " + totalVehicles);

System.***out***.println("Total revenue collected: Rs" + totalRevenue);

}

}

**public** **class** project4 {

**public** **static** **void** main(String[] args) {

TollBoothRevenueManager toll = **new** TollBoothRevenueManager();

toll.acceptRecord();

toll.setTollRates();

toll.calculateRevenue();

toll.printRecord();

}

}

