**Assignment3**

**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.**

**Program code:**

**package Cal\_Loan;**

**import java.util.Scanner;**

**public class Loan\_Cal {**

**private double principal;**

**private double annualInterestRate;**

**private int loanTerm;**

**public void acceptRecord() {**

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

**System.*out*.print("Enter the principal amount (loan amount): ₹");**

**principal = scanner.nextDouble();**

**System.*out*.print("Enter the annual interest rate (in percentage): ");**

**annualInterestRate = scanner.nextDouble();**

**System.*out*.print("Enter the loan term (in years): ");**

**loanTerm = scanner.nextInt();**

**}**

**public double calculateMonthlyPayment() {**

**double monthlyInterestRate = annualInterestRate / 12 / 100;**

**int numberOfMonths = loanTerm \* 12;**

**double numerator = monthlyInterestRate \* Math.*pow*(1 + monthlyInterestRate, numberOfMonths);**

**double denominator = Math.*pow*(1 + monthlyInterestRate, numberOfMonths) - 1;**

**return principal \* (numerator / denominator);**

**}**

**public void printRecord() {**

**double monthlyPayment = calculateMonthlyPayment();**

**double totalAmountPaid = monthlyPayment \* loanTerm \* 12;**

**System.*out*.printf("Monthly Payment: ₹%.2f%n", monthlyPayment);**

**System.*out*.printf("Total Amount Paid over the life of the loan: ₹%.2f%n", totalAmountPaid);**

**}**

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

**Loan\_Cal calculator = new Loan\_Cal();**

**calculator.acceptRecord();**

**calculator.printRecord();**

**}**

**}**

**Output:**

**Enter the principal amount (loan amount): ₹20000**

**Enter the annual interest rate (in percentage): 12**

**Enter the loan term (in years): 2024**

**Monthly Payment: ₹200.00**

**Total Amount Paid over the life of the loan: ₹4857600.00**

**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.**

**Output:**

**Enter the initial investment amount: ₹500000**

**Enter the annual interest rate (in percentage): 8**

**Enter the number of times the interest is compounded per year: 15**

**Enter the investment duration (in years): 9**

**Future Value of the investment: ₹1025253.21**

**Total Interest Earned: ₹525253.21**

**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.**

**Program code:**

**package Cal\_Loan;**

**import java.util.Scanner;**

**public class BMI\_Tracker {**

**private double weight;**

**private double height;**

**private double bmi;**

**public void acceptRecord() {**

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

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

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

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

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

**}**

**public void calculateBMI() {**

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

**}**

**public String classifyBMI() {**

**if (bmi < 18.5) {**

**return "Underweight";**

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

**return "Normal weight";**

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

**return "Overweight";**

**} else {**

**return "Obese";**

**}**

**}**

**public void printRecord() {**

**System.*out*.printf("BMI: %.2f%n", bmi);**

**System.*out*.println("Classification: " + classifyBMI());**

**}**

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

**BMI\_Tracker tracker = new BMI\_Tracker();**

**tracker.acceptRecord();**

**tracker.calculateBMI();**

**tracker.printRecord();**

**}**

**}**

**Output:**

**Enter weight (in kilograms): 45**

**Enter height (in meters): 5.3**

**BMI: 1.60**

**Classification: Underweight**

**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.**

**Program code:**

**package Cal\_Loan;**

**import java.util.Scanner;**

**public class Retail\_Sale {**

**private double originalPrice;**

**private double discountRate;**

**private double discountAmount;**

**private double finalPrice;**

**public void acceptRecord() {**

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

**System.*out*.print("Enter the original price of the item: ₹");**

**originalPrice = scanner.nextDouble();**

**System.*out*.print("Enter the discount percentage: ");**

**discountRate = scanner.nextDouble();**

**}**

**public void calculateDiscount() {**

**discountAmount = originalPrice \* (discountRate / 100);**

**finalPrice = originalPrice - discountAmount;**

**}**

**public void printRecord() {**

**System.*out*.printf("Discount Amount: ₹%.2f%n", discountAmount);**

**System.*out*.printf("Final Price: ₹%.2f%n", finalPrice);**

**}**

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

**Retail\_Sale calculator = new Retail\_Sale();**

**calculator.acceptRecord();**

**calculator.calculateDiscount();**

**calculator.printRecord();**

**}**

**}**

**Output:**

**Enter the original price of the item: ₹100**

**Enter the discount percentage: 20**

**Discount Amount: ₹20.00**

**Final Price: ₹80.00**

**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.**

**Program code:**

**package Cal\_Loan;**

**import java.util.Scanner;**

**public class Toll\_Managment {**

**private double carRate;**

**private double truckRate;**

**private double motorcycleRate;**

**private int numberOfCars;**

**private int numberOfTrucks;**

**private int numberOfMotorcycles;**

**public void setTollRates() {**

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

**System.*out*.print("Enter the toll rate for Car (₹): ");**

**carRate = scanner.nextDouble();**

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

**truckRate = scanner.nextDouble();**

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

**motorcycleRate = scanner.nextDouble();**

**}**

**public void acceptRecord() {**

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

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

**numberOfCars = scanner.nextInt();**

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

**numberOfTrucks = scanner.nextInt();**

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

**numberOfMotorcycles = scanner.nextInt();**

**}**

**public double calculateRevenue() {**

**double revenueFromCars = numberOfCars \* carRate;**

**double revenueFromTrucks = numberOfTrucks \* truckRate;**

**double revenueFromMotorcycles = numberOfMotorcycles \* motorcycleRate;**

**return revenueFromCars + revenueFromTrucks + revenueFromMotorcycles;**

**}**

**public void printRecord() {**

**int totalVehicles = numberOfCars + numberOfTrucks + numberOfMotorcycles;**

**double totalRevenue = calculateRevenue();**

**System.*out*.printf("Total Number of Vehicles: %d%n", totalVehicles);**

**System.*out*.printf("Total Revenue Collected: ₹%.2f%n", totalRevenue);**

**}**

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

**Toll\_Managment manager = new Toll\_Managment();**

**manager.setTollRates();**

**manager.acceptRecord();**

**manager.printRecord();**

**}**

**}**

**Output:**

**Enter the toll rate for Car (₹): 200**

**Enter the toll rate for Truck (₹): 100**

**Enter the toll rate for Motorcycle (₹): 50**

**Enter the number of Cars: 10**

**Enter the number of Trucks: 20**

**Enter the number of Motorcycles: 5**

**Total Number of Vehicles: 35**

**Total Revenue Collected: ₹4250.00**