**Assignmnet 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:
   * **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 (₹).

**Code :-**

**package** calculator;

**import** java.util.Scanner;

**public** **class** Program1 {

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

// Create a Scanner object to take input from the user

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

// Accept the principal amount (loan amount) from the user

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

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

// Accept the annual interest rate from the user

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

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

// Accept the loan term (in years) from the user

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

**int** loanTermYears = scanner.nextInt();

// Close the scanner

scanner.close();

// Calculate the monthly interest rate

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

// Calculate the number of months for the loan

**int** numberOfMonths = loanTermYears \* 12;

// Calculate the monthly payment using the mortgage formula

**double** monthlyPayment = principal \* (monthlyInterestRate \* Math.*pow*(1 + monthlyInterestRate, numberOfMonths))

/ (Math.*pow*(1 + monthlyInterestRate, numberOfMonths) - 1);

// Calculate the total amount paid over the life of the loan

**double** totalPayment = monthlyPayment \* numberOfMonths;

// Display the results

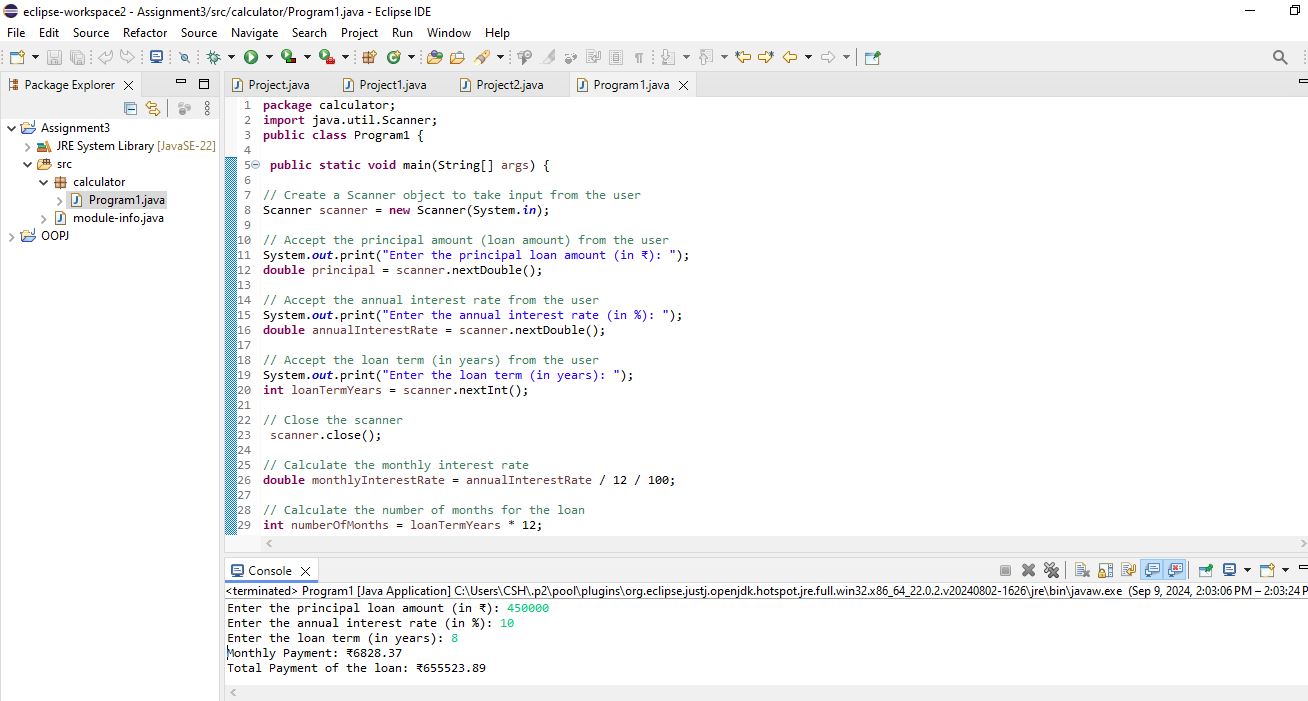
System.***out***.printf("Monthly Payment: ₹%.2f%n", monthlyPayment);

System.***out***.printf("Total Payment of the loan: ₹%.2f%n", totalPayment);

}

}

Output :-



**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 (₹).

Code :-

**package** calculator;

**import** java.util.Scanner;

**public** **class** Program2 {

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

// Create a Scanner object to take input from the user

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

// Accept the initial investment (principal) amount from the user

System.***out***.print("Enter the initial investment amount (in ₹): ");

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

// Accept the annual interest rate from the user

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

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

// Accept the number of times interest is compounded per year

System.***out***.print("Enter the number of times interest is compounded per year: ");

**int** numberOfCompounds = scanner.nextInt();

// Accept the investment duration (in years) from the user

System.***out***.print("Enter the investment duration (in years): ");

**int** years = scanner.nextInt();

// Close the scanner as we no longer need user input

scanner.close();

// Convert annual interest rate from percentage to decimal

**double** ratePerCompound = annualInterestRate / 100;

// Calculate the future value using the compound interest formula

**double** futureValue = principal \* Math.*pow*(1 + annualInterestRate / numberOfCompounds, numberOfCompounds \* years);

// Calculate the total interest earned

**double** totalInterest = futureValue - principal;

// Display the future value and total interest earned

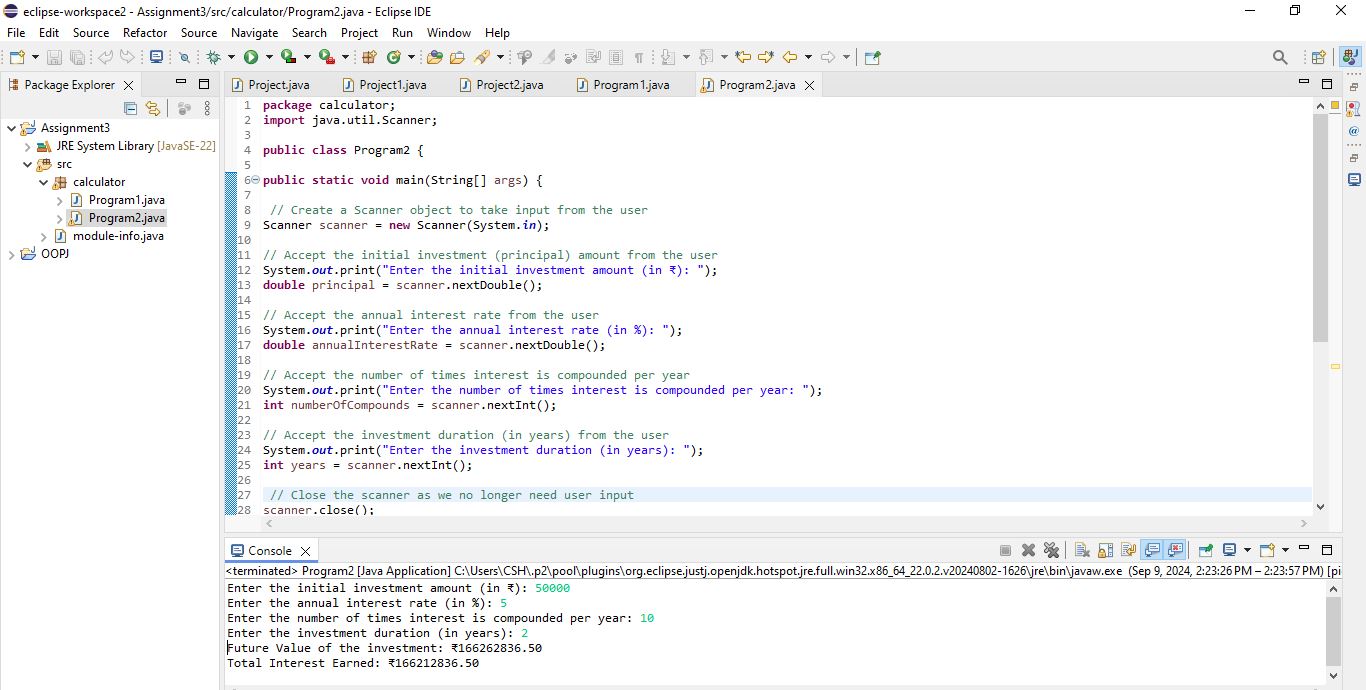
System.***out***.printf("Future Value of the investment: ₹%.2f%n", futureValue);

System.***out***.printf("Total Interest Earned: ₹%.2f%n", totalInterest);

}

}

Output :-



**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.

**Code :-**

**package** calculator;

**import** java.util.Scanner;

**public** **class** Program3 {

**public** **static** **double** calculateBMI(**double** weight, **double** height) {

**return** weight / (height \* height);

}

**public** **static** String classifyBMI(**double** bmi) {

**if** (bmi < 18.5) {

**return** "Underweight";

} **else** **if** (bmi < 24.9) {

**return** "Normal weight";

} **else** **if** (bmi < 29.9) {

**return** "Overweight";

} **else** {

**return** "Obese";

}

}

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

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

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

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

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

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

**double** bmi = *calculateBMI*(weight, height);

String classification = *classifyBMI*(bmi);

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

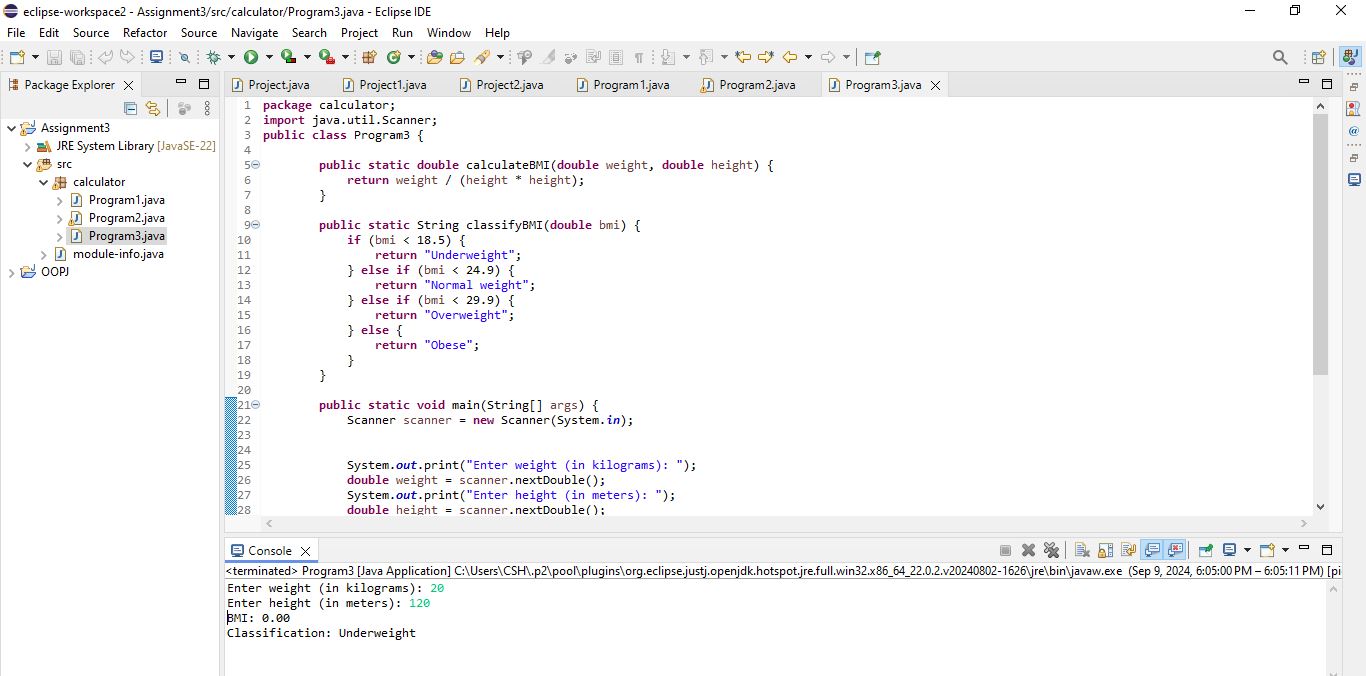
System.***out***.println("Classification: " + classification);

scanner.close();

}

}

Output :-



**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 (₹).

**Code :-**

**package** calculator;

**import** java.util.Scanner;

**public** **class** Program4 {

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

//Create a Scanner object to accept input

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

// Accept original price and discount percentage from the user

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

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

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

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

// Calculate the discount amount

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

// Calculate the final price after applying the discount

**double** finalPrice = originalPrice - discountAmount;

// Display the discount amount and final price

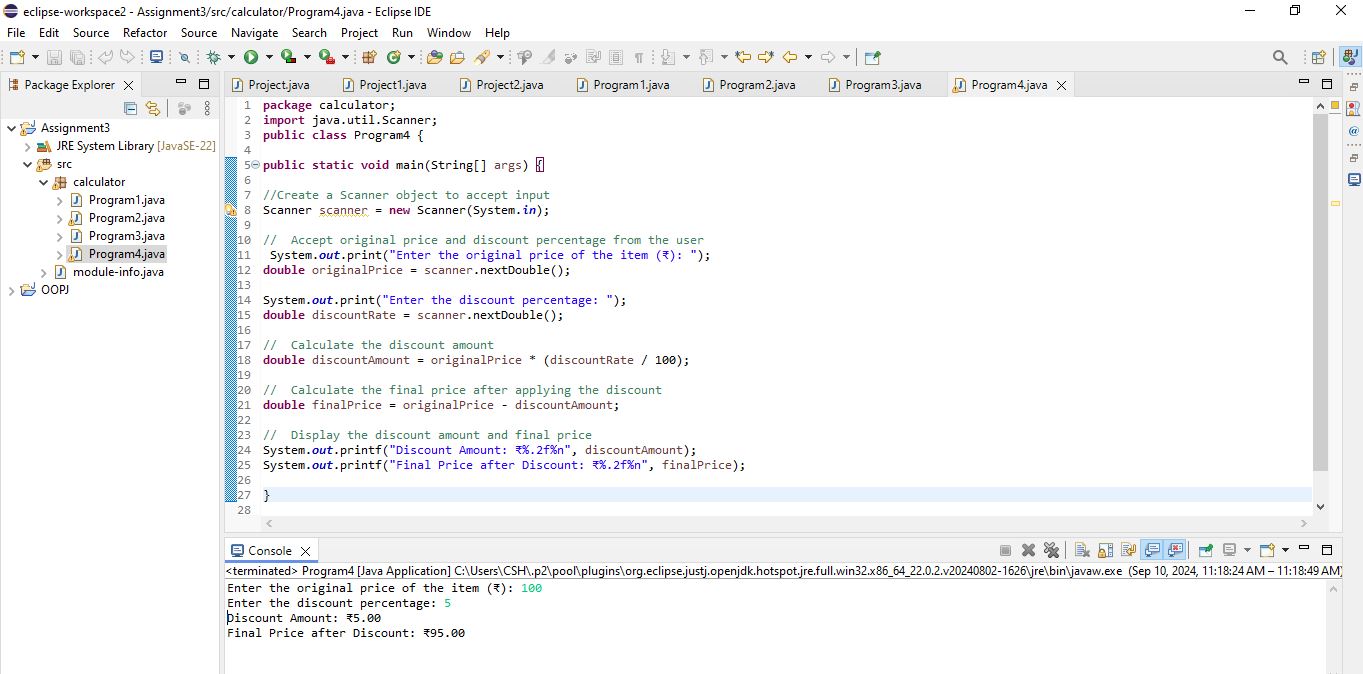
System.***out***.printf("Discount Amount: ₹%.2f%n", discountAmount);

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

}

}

**Output :-**

****

**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

Code :-

**package** calculator;

**import** java.util.Scanner;

**public** **class** Program5 {

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

//Create a Scanner object for user input

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

// Set toll rates for each vehicle type

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

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

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

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

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

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

// Accept the number of vehicles passing through

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

**int** carCount = scanner.nextInt();

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

**int** truckCount = scanner.nextInt();

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

**int** motorcycleCount = scanner.nextInt();

//Calculate total revenue for each vehicle type

**double** carRevenue = carRate \* carCount;

**double** truckRevenue = truckRate \* truckCount;

**double** motorcycleRevenue = motorcycleRate \* motorcycleCount;

// Calculate total revenue and total number of vehicles

**double** totalRevenue = carRevenue + truckRevenue + motorcycleRevenue;

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

// Display the total revenue and total number of vehicles

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

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

}

}

Output :-

