

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:
 - **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 (₹).

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 :-

The screenshot shows the Eclipse IDE with the 'Program1.java' file open. The code is as follows:

```

1 package calculator;
2 import java.util.Scanner;
3 public class Program1 {
4
5     public static void main(String[] args) {
6
7         // Create a Scanner object to take input from the user
8         Scanner scanner = new Scanner(System.in);
9
10        // Accept the principal amount (loan amount) from the user
11        System.out.print("Enter the principal loan amount (in ₹): ");
12        double principal = scanner.nextDouble();
13
14        // Accept the annual interest rate from the user
15        System.out.print("Enter the annual interest rate (in %): ");
16        double annualInterestRate = scanner.nextDouble();
17
18        // Accept the loan term (in years) from the user
19        System.out.print("Enter the loan term (in years): ");
20        int loanTermYears = scanner.nextInt();
21
22        // Close the scanner
23        scanner.close();
24
25        // Calculate the monthly interest rate
26        double monthlyInterestRate = annualInterestRate / 12 / 100;
27
28        // Calculate the number of months for the loan
29        int numberOfMonths = loanTermYears * 12;
30    }
31 }

```

The console output shows the program execution with the following input and output:

```

C:\Users\CSH.p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1628\jre\bin\javaw.exe (Sep 9, 2024 2:03:06 PM - 2:03:24 P
Enter the principal loan amount (in ₹): 450000
Enter the annual interest rate (in %): 10
Enter the loan term (in years): 8
Monthly Payment: ₹6828.37
Total Payment of the loan: ₹655523.89

```

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:
 - o **Future Value Calculation:**

$$\text{futureValue} = \text{principal} * (1 + \text{annualInterestRate} / \text{numberOfCompounds}) ^ (\text{numberOfCompounds} * \text{years})$$
 - o **Total Interest Earned:** $\text{totalInterest} = \text{futureValue} - \text{principal}$
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();

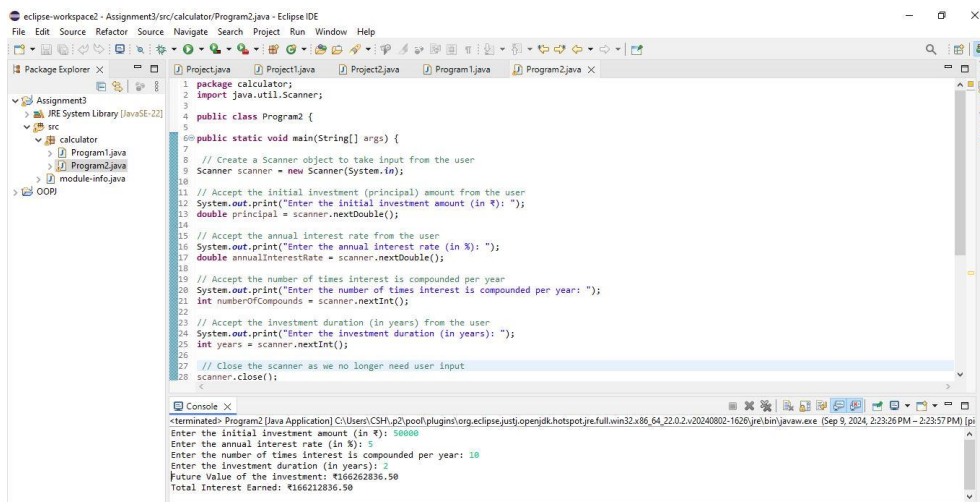
        // Convert annual interest rate from percentage to decimal
        double ratePerCompound = annualInterestRate / 100;

        // 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 :-



```
1 package calculator;
2 import java.util.Scanner;
3
4 public class Program2 {
5
6     public static void main(String[] args) {
7
8         // Create a Scanner object to take input from the user
9         Scanner scanner = new Scanner(System.in);
10
11         // Accept the initial investment (principal) amount from the user
12         System.out.print("Enter the initial investment amount (in ₹): ");
13         double principal = scanner.nextDouble();
14
15         // Accept the annual interest rate from the user
16         System.out.print("Enter the annual interest rate (in %): ");
17         double annualInterestRate = scanner.nextDouble();
18
19         // Accept the number of times interest is compounded per year
20         System.out.print("Enter the number of times interest is compounded per year: ");
21         int numberOfCompounds = scanner.nextInt();
22
23         // Accept the investment duration (in years) from the user
24         System.out.print("Enter the investment duration (in years): ");
25         int years = scanner.nextInt();
26
27         // Close the scanner as we no longer need user input
28         scanner.close();
29
30         // Calculate the future value of the investment
31         double futureValue = calculateFutureValue(principal, annualInterestRate, numberOfCompounds, years);
32
33         // Display the future value and total interest earned
34         System.out.println("Future Value of the investment: ₹" + futureValue);
35         System.out.println("Total Interest Earned: ₹" + (futureValue - principal));
36     }
37
38     // Method to calculate the future value of an investment
39     public static double calculateFutureValue(double principal, double annualInterestRate, int numberOfCompounds, int years) {
40         double futureValue = principal * Math.pow(1 + (annualInterestRate / 100) / numberOfCompounds, numberOfCompounds * years);
41         return futureValue;
42     }
43 }
```

```
<terminated>- Program2 [Java Application] C:\Users\poo\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0.2.v20240802-1620\jre\bin\javaw.exe (Sep 9, 2024, 2:23:26 PM - 2:23:57 PM) [pi]
Enter the initial investment amount (in ₹): 50000
Enter the annual interest rate (in %): 5
Enter the number of times interest is compounded per year: 10
Enter the investment duration (in years): 2
Future Value of the investment: ₹166262836.50
Total Interest Earned: ₹166212836.50
```

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.

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";
        }
    }
}
```

```

        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 :-

```

1 package calculator;
2 import java.util.Scanner;
3 public class Program3 {
4
5     public static double calculateBMI(double weight, double height) {
6         return weight / (height * height);
7     }
8
9     public static String classifyBMI(double bmi) {
10        if (bmi < 18.5) {
11            return "Underweight";
12        } else if (bmi < 24.9) {
13            return "Normal weight";
14        } else if (bmi < 29.9) {
15            return "Overweight";
16        } else {
17            return "Obese";
18        }
19    }
20
21    public static void main(String[] args) {
22        Scanner scanner = new Scanner(System.in);
23
24        System.out.print("Enter weight (in kilograms): ");
25        double weight = scanner.nextDouble();
26        System.out.print("Enter height (in meters): ");
27        double height = scanner.nextDouble();
28    }
29 }

```

```

<terminated> Program3 [Java Application] C:\Users\p2\p2\p2\plugin\org.eclipse.jdt.launcher\org.eclipse.jdt.launcher.exe (Sep 9, 2024, 6:05:00 PM - 6:05:11 PM) [p
Enter weight (in kilograms): 20
Enter height (in meters): 120
BMI: 0.00
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:** $\text{discountAmount} = \text{originalPrice} * (\text{discountRate} / 100)$
 - **Final Price Calculation:** $\text{finalPrice} = \text{originalPrice} - \text{discountAmount}$
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);

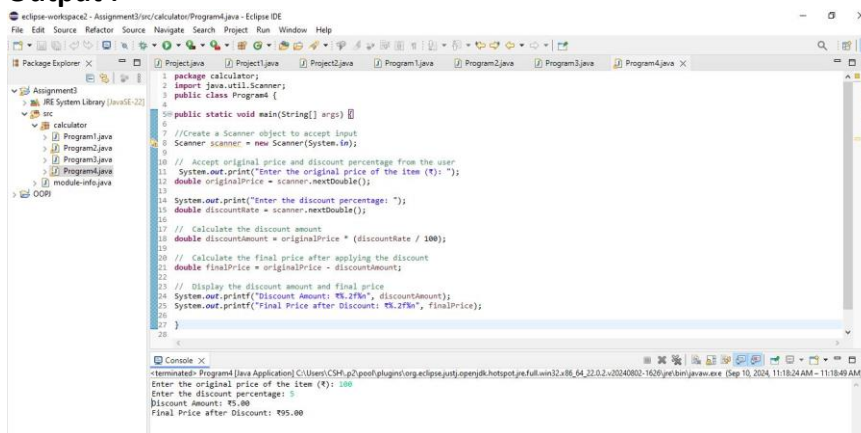
        // Calculate the final price after applying the discount
        double finalPrice = originalPrice - discountAmount;

        // 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;

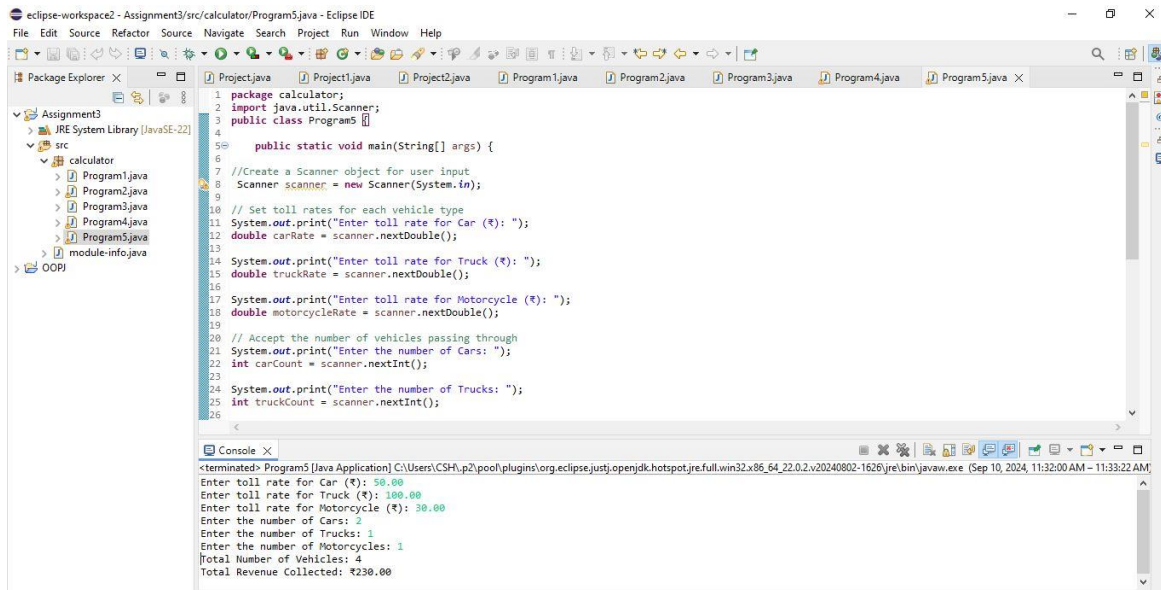
// Display the total revenue and total number of vehicles
System.out.printf("Total Number of Vehicles: %d\n", totalVehicles);
System.out.printf("Total Revenue Collected: ₹%.2f\n", totalRevenue);

}

}

```

Output :-



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Assignment3' with a sub-package 'calculator' containing five Java files: Program1.java, Program2.java, Program3.java, Program4.java, and Program5.java. The main editor displays the source code of Program5.java, which is a Java application that calculates the total revenue and total number of vehicles based on user input. The code includes comments and uses the Scanner class for input and printf for formatted output. The Console window at the bottom shows the execution output, which matches the expected results from the provided code.

```

<terminated> Program5 [Java Application] C:\Users\CSH\p2\pooth\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (Sep 10, 2024, 11:32:00 AM - 11:33:22 AM)
Enter toll rate for Car (₹): 50.00
Enter toll rate for Truck (₹): 100.00
Enter toll rate for Motorcycle (₹): 30.00
Enter the number of Cars: 2
Enter the number of Trucks: 1
Enter the number of Motorcycles: 1
Total Number of Vehicles: 4
Total Revenue Collected: ₹230.00

```