

ASSIGNMENT NO. :- 4

Kunal More

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 (₹).

Define the class `LoanAmortizationCalculator` with fields, an appropriate constructor, getter and setter methods, a `toString` method and business logic methods. Define the class `LoanAmortizationCalculatorUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method and test the functionality of the utility class.

P1.java

```
package four.assignment.sir;

public class p1 {
    public static void main(String[] args) {
        util util = new util();
        util.menuList();
    }
}
```

Lac.java

```
package four.assignment.sir;

import java.util.Scanner;

public class lac {

    private lac calculator;
```

ASSIGNMENT NO. :- 4

```
private double principal; // principal amount
private double air; // annual interest rate
private double lty; // loan term in years

public lac(double principal ,double air ,double lty)
{
    this.principal = principal;
    this.air = air;
    this.lty = lty;
}

public lac getCalculator() {
    return calculator;
}

public void setCalculator(lac calculator) {
    this.calculator = calculator;
}

public double getPrincipal() {
    return principal;
}

public void setPrincipal(double principal) {
    this.principal = principal;
}

public double getAir() {
    return air;
}

public void setAir(double air) {
    this.air = air;
}

public double getLty() {
    return lty;
}
```

ASSIGNMENT NO. :- 4

```
public void setLty(double lty) {
    this.lty = lty;
}

public double calculateMonthlyPayment() {
    double monthlyInterestRate = (air / 12) / 100;
    int numberOfMonths = (int) (lty * 12);
    return principal * (monthlyInterestRate *
Math.pow(1 + monthlyInterestRate, numberOfMonths))
        / (Math.pow(1 + monthlyInterestRate,
numberOfMonths) - 1);
}

public double calculateTotalPayment() {
    return calculateMonthlyPayment() * lty * 12;
}

public String toString() {
    return "Principal: ₹" + principal + "\nAnnual
Interest Rate: " + lty + "%\nLoan Term: " + lty + "
years";
}

public void menuList() {
    // TODO Auto-generated method stub
}

}
```

Util.java

```
package four.assignment.sir;

import java.util.Scanner;

public class util {
    private lac loan;

    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);
```

ASSIGNMENT NO. :- 4

```
        System.out.println("Enter the Principal Amount
(₹): ");
        double principal = sc.nextDouble();

        System.out.println("Enter the Annual Interest Rate
(%): ");
        double air = sc.nextDouble();

        System.out.println("Enter Loan Term (in years):
");
        double lty = sc.nextDouble();

        loan = new lac(principal, air, lty);
    }

    public void printRecord() {
        if (loan != null) {
            System.out.println("\nLoan Details:");
            System.out.println(loan);
            double monthlyPayment =
loan.calculateMonthlyPayment();
            double totalPayment =
loan.calculateTotalPayment();
            System.out.println("Monthly Payment: ₹" +
String.format("%.2f", monthlyPayment));
            System.out.println("Total Payment over Loan
Term: ₹" + String.format("%.2f", totalPayment));
        } else {
            System.out.println("No loan data available.
Please input the loan details first.");
        }
    }

    public void menuList() {
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n--- Loan Amortization
Calculator Menu ---");
            System.out.println("1. Enter Loan Details");
```

ASSIGNMENT NO. :- 4

```
        System.out.println("2. Display Loan Details  
and Payment");  
        System.out.println("0. Exit");  
        System.out.print("Enter your choice: ");  
        choice = sc.nextInt();  
  
        switch (choice) {  
            case 1:  
                acceptRecord();  
                break;  
            case 2:  
                printRecord();  
                break;  
            case 0:  
                System.out.println("Exiting...");  
                break;  
            default:  
                System.out.println("Invalid choice.  
Please try again.");  
        }  
    } while (choice != 0);  
}
```

```
--- Loan Amortization Calculator Menu ---  
1. Enter Loan Details  
2. Display Loan Details and Payment  
0. Exit  
Enter your choice: 1  
Enter the Principal Amount (₹):  
5000  
Enter the Annual Interest Rate (%):  
4  
Enter Loan Term (in years):  
5  
  
--- Loan Amortization Calculator Menu ---  
1. Enter Loan Details  
2. Display Loan Details and Payment  
0. Exit  
Enter your choice: 2  
  
Loan Details:  
Principal: ₹5000.0  
Annual Interest Rate: 5.0%  
Loan Term: 5.0 years  
Monthly Payment: ₹92.08  
Total Payment over Loan Term: ₹5524.96
```

ASSIGNMENT NO. :- 4

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 (₹).

Define the class `CompoundInterestCalculator` with fields, an appropriate constructor, getter and setter methods, a `toString` method and business logic methods. Define the class `CompoundInterestCalculatorUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method to test the functionality of the utility class.

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:** $\text{BMI} = \text{weight} / (\text{height} * \text{height})$
3. Classify the BMI into one of the following categories:
 - o Underweight: $\text{BMI} < 18.5$
 - o Normal weight: $18.5 \leq \text{BMI} < 24.9$
 - o Overweight: $25 \leq \text{BMI} < 29.9$
 - o Obese: $\text{BMI} \geq 30$
4. Display the BMI value and its classification.

Define the class `BMITracker` with fields, an appropriate constructor, getter and setter methods, a `toString` method, and business logic methods. Define the class `BMITrackerUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method to test the functionality of the utility class.

P3.java

```
package four.assignment.sir;

public class p3 {
    public static void main(String a[]) {
        BTU util = new BTU();
        util.menuList();
    }
}
```

ASSIGNMENT NO. :- 4

```
}  
}
```

BTU.java

```
package four.assignment.sir;  
  
import java.util.Scanner;  
  
public class BTU {  
    private BMITracker tracker;  
  
    // Method to accept input from the user  
    public void acceptRecord() {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter Weight (in kg): ");  
        double weight = sc.nextDouble();  
  
        System.out.print("Enter Height (in meters): ");  
        double height = sc.nextDouble();  
  
        tracker = new BMITracker(weight, height);  
    }  
  
    // Method to display the BMI value and its  
    classification  
    public void printRecord() {  
        if (tracker != null) {  
            System.out.println("\nBMI Details:");  
            System.out.println(tracker);  
            double bmi = tracker.calculateBMI();  
            String classification = tracker.classifyBMI();  
            System.out.println("BMI: " +  
String.format("%.2f", bmi));  
            System.out.println("BMI Classification: " +  
classification);  
        } else {  
            System.out.println("No BMI data available.  
Please input the details first.");  
        }  
    }  
}
```

ASSIGNMENT NO. :- 4

```
}

// Menu method
public void menuList() {
    Scanner sc = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\n--- BMI Tracker Menu ---");

        System.out.println("1. Enter BMI Details");
        System.out.println("2. Display BMI Details and Classification");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = sc.nextInt();

        switch (choice) {
            case 1:
                acceptRecord();
                break;
            case 2:
                printRecord();
                break;
            case 0:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 0);
}
}
```

BMITracker.java

```
package four.assignment.sir;

public class BMITracker {
    private double weight; // Weight in kilograms
```


ASSIGNMENT NO. :- 4

```
private double height; // Height in meters

// Constructor
public BMITracker(double weight, double height) {
    this.weight = weight;
    this.height = height;
}

// Getters and Setters
public double getWeight() {
    return weight;
}

public void setWeight(double weight) {
    this.weight = weight;
}

public double getHeight() {
    return height;
}

public void setHeight(double height) {
    this.height = height;
}

// Method to calculate BMI
public double calculateBMI() {
    return weight / (height * height);
}

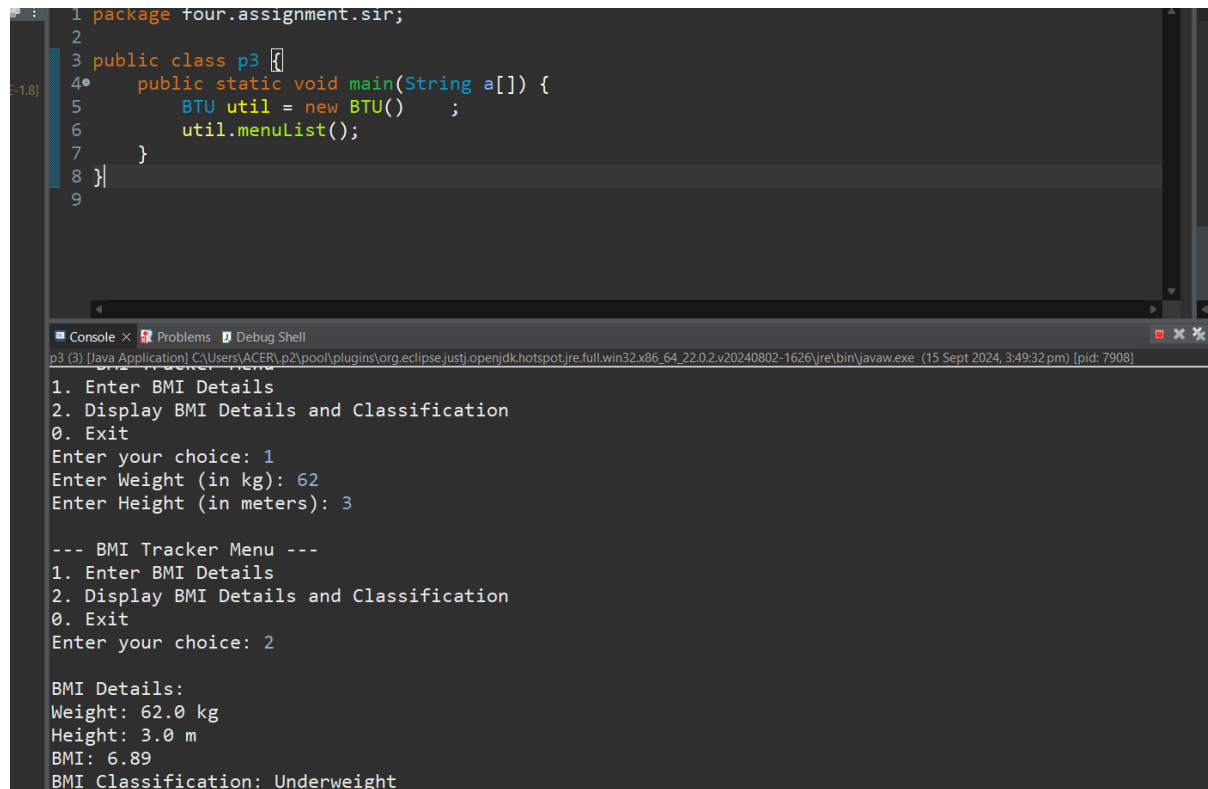
// Method to classify the BMI
public String classifyBMI() {
    double bmi = calculateBMI();
    if (bmi < 18.5) {
        return "Underweight";
    } else if (bmi >= 18.5 && bmi < 24.9) {
        return "Normal weight";
    } else if (bmi >= 25 && bmi < 29.9) {
        return "Overweight";
    } else {
        return "Obese";
    }
}
```

ASSIGNMENT NO. :- 4

```
}

@Override
public String toString() {
    return "Weight: " + weight + " kg\nHeight: " +
height + " m";
}
}
```

OUTPUT > >



The screenshot shows an IDE with a Java file named `p3` in the package `four.assignment.sir`. The code defines a `main` method that creates a `BTU` object and calls `menuList()`. The console output shows the program running, displaying a menu with options 1 (Enter BMI Details), 2 (Display BMI Details and Classification), and 0 (Exit). The user enters choice 1, then weight 62 and height 3. The program displays the BMI details: Weight: 62.0 kg, Height: 3.0 m, BMI: 6.89, and BMI Classification: Underweight. The user then enters choice 2, and the program displays the BMI Tracker Menu again.

```
1 package four.assignment.sir;
2
3 public class p3 {
4     public static void main(String a[]) {
5         BTU util = new BTU();
6         util.menuList();
7     }
8 }
9
```

Console Output:

```
p3 (3) [Java Application] C:\Users\ACER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (15 Sept 2024, 3:49:32 pm) [pid: 7908]
1. Enter BMI Details
2. Display BMI Details and Classification
0. Exit
Enter your choice: 1
Enter Weight (in kg): 62
Enter Height (in meters): 3

--- BMI Tracker Menu ---
1. Enter BMI Details
2. Display BMI Details and Classification
0. Exit
Enter your choice: 2

BMI Details:
Weight: 62.0 kg
Height: 3.0 m
BMI: 6.89
BMI 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:
 - o **Discount Amount Calculation:** $\text{discountAmount} = \text{originalPrice} * (\text{discountRate} / 100)$
 - o **Final Price Calculation:** $\text{finalPrice} = \text{originalPrice} - \text{discountAmount}$
3. Display the discount amount and the final price of the item, in Indian Rupees (₹).

ASSIGNMENT NO. :- 4

Define the class `DiscountCalculator` with fields, an appropriate constructor, getter and setter methods, a `toString` method, and business logic methods. Define the class `DiscountCalculatorUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method to test the functionality of the utility class.

P4.java

```
package four.assignment.sir;

public class p4 {
    public static void main(String[] args) {
        DiscountCalculatorUtil util = new
DiscountCalculatorUtil();
        util.menuList();
    }
}
```

DiscountCalculatorUtil.java

```
package four.assignment.sir;

import java.util.Scanner;

public class DiscountCalculatorUtil {
    private DiscountCalculator calculator;

    // Method to accept input from the user
    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Original Price (₹):
");
        double originalPrice = sc.nextDouble();

        System.out.print("Enter the Discount Rate (%): ");
        double discountRate = sc.nextDouble();

        calculator = new DiscountCalculator(originalPrice,
discountRate);
    }
}
```

ASSIGNMENT NO. :- 4

```
// Method to display the discount details and final price
public void printRecord() {
    if (calculator != null) {
        System.out.println("\nDiscount Details:");
        System.out.println(calculator);
        double discountAmount =
calculator.calculateDiscountAmount();
        double finalPrice =
calculator.calculateFinalPrice();
        System.out.println("Discount Amount: ₹" +
String.format("%.2f", discountAmount));
        System.out.println("Final Price: ₹" +
String.format("%.2f", finalPrice));
    } else {
        System.out.println("No discount data
available. Please input the item details first.");
    }
}

// Menu method
public void menuList() {
    Scanner sc = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\n--- Discount Calculator
Menu ---");
        System.out.println("1. Enter Item Details");
        System.out.println("2. Display Discount
Details and Final Price");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = sc.nextInt();

        switch (choice) {
            case 1:
                acceptRecord();
                break;
            case 2:
                printRecord();
                break;
        }
    }
}
```

ASSIGNMENT NO. :- 4

```
        case 0:
            System.out.println("Exiting...");
            break;
        default:
            System.out.println("Invalid choice.
Please try again.");
    }
} while (choice != 0);
}
}
```

DiscountCalculator.java

```
package four.assignment.sir;

public class DiscountCalculator {
    private double originalPrice;
    private double discountRate;

    public DiscountCalculator(double originalPrice, double
discountRate) {
        this.originalPrice = originalPrice;
        this.discountRate = discountRate;
    }

    public double getOriginalPrice() {
        return originalPrice;
    }

    public void setOriginalPrice(double originalPrice) {
        this.originalPrice = originalPrice;
    }

    public double getDiscountRate() {
        return discountRate;
    }

    public void setDiscountRate(double discountRate) {
        this.discountRate = discountRate;
    }

    public double calculateDiscountAmount() {
```

ASSIGNMENT NO. :- 4

```
        return originalPrice * (discountRate / 100);
    }

    public double calculateFinalPrice() {
        return originalPrice - calculateDiscountAmount();
    }

    @Override
    public String toString() {
        return "Original Price: ₹" + originalPrice +
"\nDiscount Rate: " + discountRate + "%";
    }
}
```

```
3
4 public class p4 {
5     public static void main(String[] args) {
6         DiscountCalculatorUtil util = new DiscountCalculatorUtil();
7         util.menuList();
8     }
9 }
10
```

Console × Problems Debug Shell

p4 (1) [Java Application] C:\Users\ACER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (15 Sept 2024, 4:29)

2. Display Discount Details and Final Price

0. Exit

Enter your choice: 1

Enter the Original Price (₹): 50000

Enter the Discount Rate (%): 5

--- Discount Calculator Menu ---

1. Enter Item Details

2. Display Discount Details and Final Price

0. Exit

Enter your choice: 2

Discount Details:

Original Price: ₹50000.0

Discount Rate: 5.0%

Discount Amount: ₹2500.00

Final Price: ₹47500.00

5. Toll Booth Revenue Management

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

ASSIGNMENT NO. :- 4

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 the class `TollBoothRevenueManager` with fields, an appropriate constructor, getter and setter methods, a `toString` method, and business logic methods. Define the class `TollBoothRevenueManagerUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method to test the functionality of the utility class.

P5. Java

```
package four.assignment.sir;

public class p5 {
    public static void main(String[] args) {
        TollBoothRevenueManagerUtil util = new
TollBoothRevenueManagerUtil();
        util.menuList();
    }
}
```

TollBoothRevenueManagerUtil.java

```
package four.assignment.sir;

import java.util.Scanner;

public class TollBoothRevenueManagerUtil {
    private TollBoothRevenueManager tollBoothManager;

    // Method to accept toll rates and number of vehicles
    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the toll rate for Cars
(₹): ");
        double carRate = sc.nextDouble();
    }
}
```

ASSIGNMENT NO. :- 4

```
        System.out.print("Enter the toll rate for Trucks
(₹): ");
        double truckRate = sc.nextDouble();

        System.out.print("Enter the toll rate for
Motorcycles (₹): ");
        double motorcycleRate = sc.nextDouble();

        tollBoothManager = new
TollBoothRevenueManager(carRate, truckRate,
motorcycleRate);

        System.out.print("\nEnter the number of Cars
passed through: ");
        tollBoothManager.setNumberOfCars(sc.nextInt());

        System.out.print("Enter the number of Trucks
passed through: ");
        tollBoothManager.setNumberOfTrucks(sc.nextInt());

        System.out.print("Enter the number of Motorcycles
passed through: ");
        tollBoothManager.setNumberOfMotorcycles(sc.nextInt());
    }

    // Method to display the toll booth details and total
    revenue
    public void printRecord() {
        if (tollBoothManager != null) {
            System.out.println("\nToll Booth Details:");
            System.out.println(tollBoothManager);
            System.out.println("Total Vehicles: " +
tollBoothManager.calculateTotalVehicles());
            System.out.println("Total Revenue Collected:
₹" + String.format("%.2f",
tollBoothManager.calculateTotalRevenue()));
        } else {
            System.out.println("No toll booth data
available. Please input the details first.");
        }
    }
}
```


ASSIGNMENT NO. :- 4

```
}

// Menu method
public void menuList() {
    Scanner sc = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\n--- Toll Booth Revenue
Manager Menu ---");
        System.out.println("1. Enter Toll Rates and
Vehicle Data");
        System.out.println("2. Display Toll Booth
Revenue and Vehicle Count");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = sc.nextInt();

        switch (choice) {
            case 1:
                acceptRecord();
                break;
            case 2:
                printRecord();
                break;
            case 0:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice.
Please try again.");
        }
    } while (choice != 0);
}
}
```

TollBoothRevenueManager.java

package four.assignment.sir;

public class TollBoothRevenueManager {

ASSIGNMENT NO. :- 4

```
private double carTollRate;
private double truckTollRate;
private double motorcycleTollRate;

private int numberOfCars;
private int numberOfTrucks;
private int numberOfMotorcycles;

// Constructor to initialize toll rates
public TollBoothRevenueManager(double carTollRate, double truckTollRate, double
motorcycleTollRate) {
    this.carTollRate = carTollRate;
    this.truckTollRate = truckTollRate;
    this.motorcycleTollRate = motorcycleTollRate;
}

// Getters and Setters for vehicle counts and toll rates
public double getCarTollRate() {
    return carTollRate;
}

public void setCarTollRate(double carTollRate) {
    this.carTollRate = carTollRate;
}

public double getTruckTollRate() {
    return truckTollRate;
}

public void setTruckTollRate(double truckTollRate) {
    this.truckTollRate = truckTollRate;
}

public double getMotorcycleTollRate() {
    return motorcycleTollRate;
}

public void setMotorcycleTollRate(double motorcycleTollRate) {
    this.motorcycleTollRate = motorcycleTollRate;
}

public int getNumberOfCars() {
    return numberOfCars;
}

public void setNumberOfCars(int numberOfCars) {
    this.numberOfCars = numberOfCars;
}
```

ASSIGNMENT NO. :- 4

```
}

public int getNumberOfTrucks() {
    return numberOfTrucks;
}

public void setNumberOfTrucks(int numberOfTrucks) {
    this.numberOfTrucks = numberOfTrucks;
}

public int getNumberOfMotorcycles() {
    return numberOfMotorcycles;
}

public void setNumberOfMotorcycles(int numberOfMotorcycles) {
    this.numberOfMotorcycles = numberOfMotorcycles;
}

// Method to calculate total revenue
public double calculateTotalRevenue() {
    return (carTollRate * numberOfCars) + (truckTollRate * numberOfTrucks) +
(motorcycleTollRate * numberOfMotorcycles);
}

// Method to calculate total number of vehicles
public int calculateTotalVehicles() {
    return numberOfCars + numberOfTrucks + numberOfMotorcycles;
}

@Override
public String toString() {
    return "Toll Rates:\nCar: ₹" + carTollRate + "\nTruck: ₹" + truckTollRate +
"\nMotorcycle: ₹" + motorcycleTollRate;
}
}
```

ASSIGNMENT NO. :- 4

```
TollBoothRevenueManagerUtil.java  p5.java  TollBoothRevenueManager.java
1 package four.assignment.sir;
2
3 public class p5 {
4     public static void main(String[] args) {
5         TollBoothRevenueManagerUtil util = new TollBoothRevenueManagerUtil();
6         util.menuList();
7     }
8 }
9

Console  Problems  Debug Shell
p5 (2) [Java Application] C:\Users\ACER\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0.2.v20240802-1626\jre\bin\javaw.exe (15 Sept 2024, 4:57:28 pm) [pid: 9296]
Enter your choice: 1
Enter the toll rate for Cars (₹): 45
Enter the toll rate for Trucks (₹): 60
Enter the toll rate for Motorcycles (₹): 90

Enter the number of Cars passed through: 45
Enter the number of Trucks passed through: 23
Enter the number of Motorcycles passed through: 22

--- Toll Booth Revenue Manager Menu ---
1. Enter Toll Rates and Vehicle Data
2. Display Toll Booth Revenue and Vehicle Count
0. Exit
Enter your choice: 2

Toll Booth Details:
Toll Rates:
Car: ₹45.0
Truck: ₹60.0
Motorcycle: ₹90.0
Total Vehicles: 90
Total Revenue Collected: ₹5385.00
```