ASSIGNMENT NO.- 3

Mayur Pawar

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 (₹).

Program.java

```
package four.assignment;

import java.util.Scanner;

public class p1 {
	public static void main(String a[]) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the Principal Amount");
		double principal = sc.nextDouble();

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

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


		LoanCalculator calculator = new LoanCalculator(principal , ainterest , years);
		calculator.display();
	}
}
```

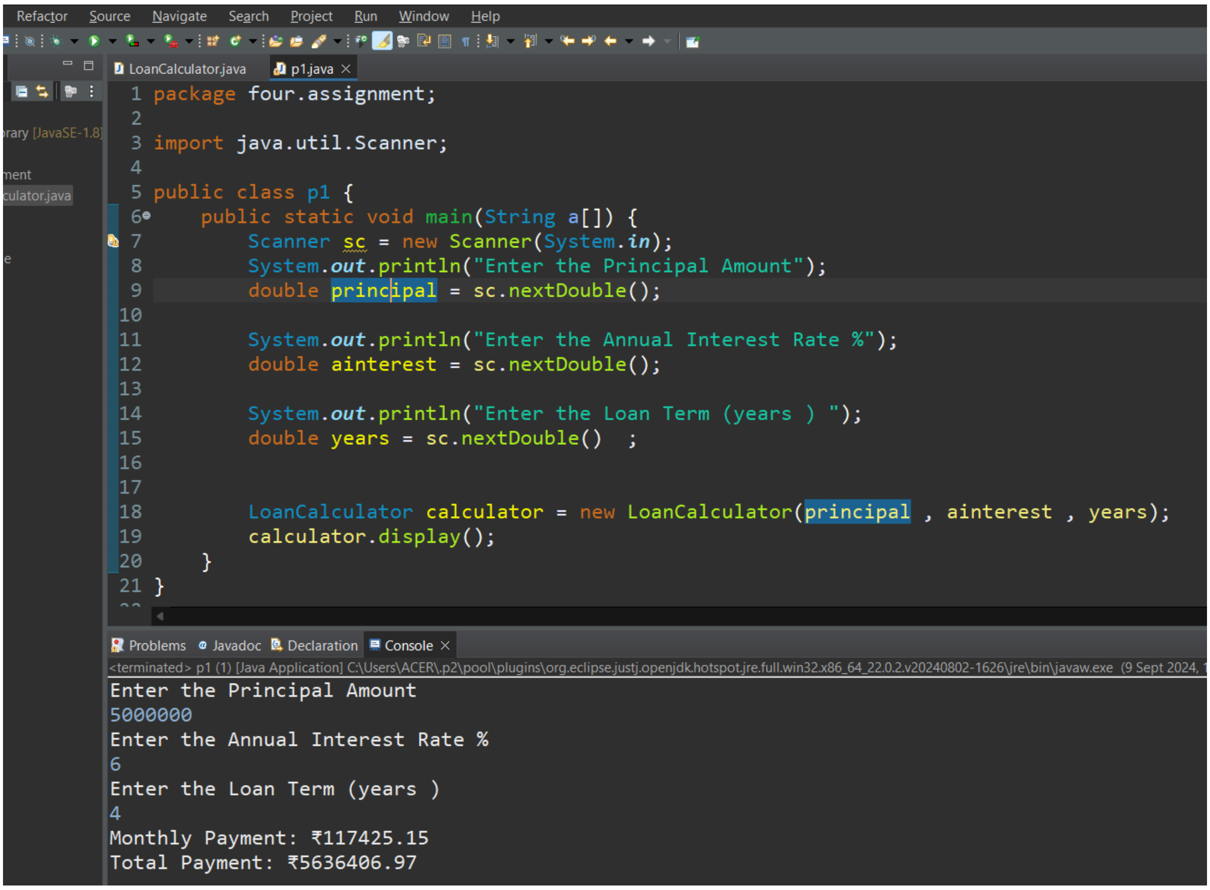
Calculator.java

```
package four.assignment;

public class LoanCalculator {

	private double years;
```

ASSIGNMENT NO.- 3

```
     private double ainterest;
     private double principal;

     public LoanCalculator(double principal, double
ainterest, double years) {
          // TODO Auto-generated constructor stub
          this.principal =principal ;
          this.ainterest = ainterest ;
          this.years = years ;

     }
     public double calculateMonthlyPayment() {
         double monthlyInterestRate = (ainterest / 12) /
100;
         int numberOfMonths = (int) (years * 12);
         double monthlyPayment = principal *
(monthlyInterestRate * Math.pow(1 + monthlyInterestRate,
numberOfMonths))
                 / (Math.pow(1 + monthlyInterestRate,
numberOfMonths) - 1);
         return monthlyPayment;
     }
     public void display() {
         double monthlyPayment = calculateMonthlyPayment();
         double totalPayment = monthlyPayment * years * 12;
         System.out.printf("Monthly Payment: ₹%.2f%n",
monthlyPayment);
         System.out.printf("Total Payment: ₹%.2f%n",
totalPayment);
     }

}
```

ASSIGNMENT NO.- 3



```java
package four.assignment;

import java.util.Scanner;

public class p1 {
    public static void main(String a[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Principal Amount");
        double principal = sc.nextDouble();

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

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


        LoanCalculator calculator = new LoanCalculator(principal , ainterest , years);
        calculator.display();
    }
}
```

<terminated> p1 (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 (9 Sept 2024,

```
Enter the Principal Amount
5000000
Enter the Annual Interest Rate %
6
Enter the Loan Term (years )
4
Monthly Payment: ₹117425.15
Total Payment: ₹5636406.97
```

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 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.

P3.java

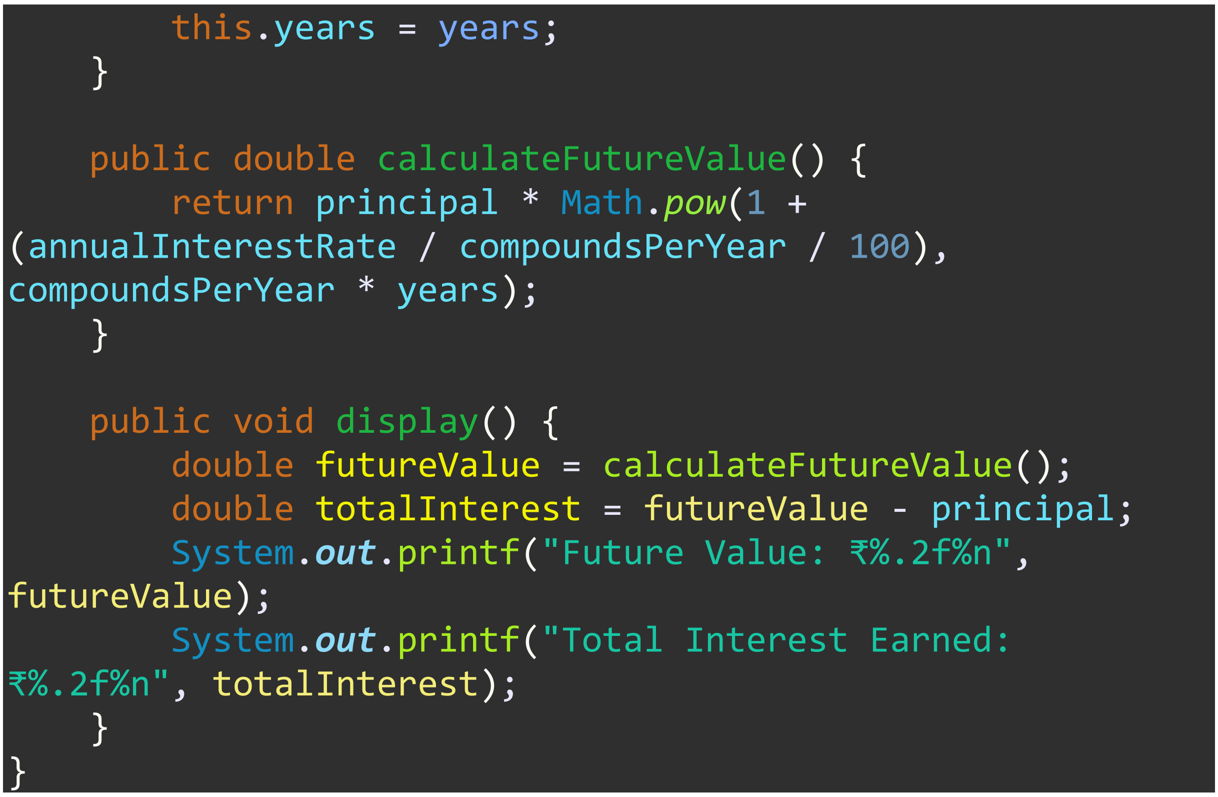
```java
package four.assignment;
```

ASSIGNMENT NO.- 3

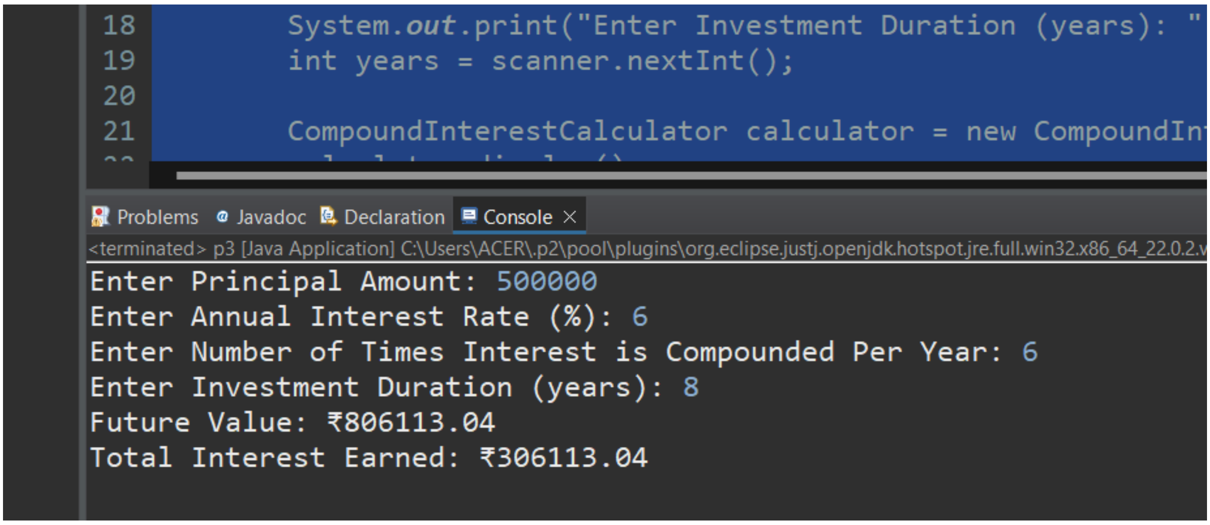
```java
import java.util.Scanner;

public class p3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Principal Amount: ");
        double principal = scanner.nextDouble();

        System.out.print("Enter Annual Interest Rate (%): 
");
        double annualInterestRate = scanner.nextDouble();

        System.out.print("Enter Number of Times Interest 
is Compounded Per Year: ");
        int compoundsPerYear = scanner.nextInt();

        System.out.print("Enter Investment Duration 
(years): ");
        int years = scanner.nextInt();

        CompoundInterestCalculator calculator = new 
CompoundInterestCalculator(principal, annualInterestRate, 
compoundsPerYear, years);
        calculator.display();
    }
}
```

CompoundInterestCalculator.java

```java
package four.assignment;
public class CompoundInterestCalculator {
    private double principal;
    private double annualInterestRate;
    private int compoundsPerYear;
    private int years;

    public CompoundInterestCalculator(double principal, 
double annualInterestRate, int compoundsPerYear, int 
years) {
        this.principal = principal;
        this.annualInterestRate = annualInterestRate;
        this.compoundsPerYear = compoundsPerYear;
```



```
        this.years = years;
    }

    public double calculateFutureValue() {
        return principal * Math.pow(1 + 
(annualInterestRate / compoundsPerYear / 100), 
compoundsPerYear * years);
    }

    public void display() {
        double futureValue = calculateFutureValue();
        double totalInterest = futureValue - principal;
        System.out.printf("Future Value: ₹%.2f%n", 
futureValue);
        System.out.printf("Total Interest Earned: 
₹%.2f%n", totalInterest);
    }
}
```



```
        System.out.print("Enter Investment Duration (years): "
        int years = scanner.nextInt();

        CompoundInterestCalculator calculator = new CompoundIn
```

```
<terminated> p3 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v
Enter Principal Amount: 500000
Enter Annual Interest Rate (%): 6
Enter Number of Times Interest is Compounded Per Year: 6
Enter Investment Duration (years): 8
Future Value: ₹806113.04
Total Interest Earned: ₹306113.04
```

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

Assignment No.- 3

   - Obese: BMI ≥ 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.

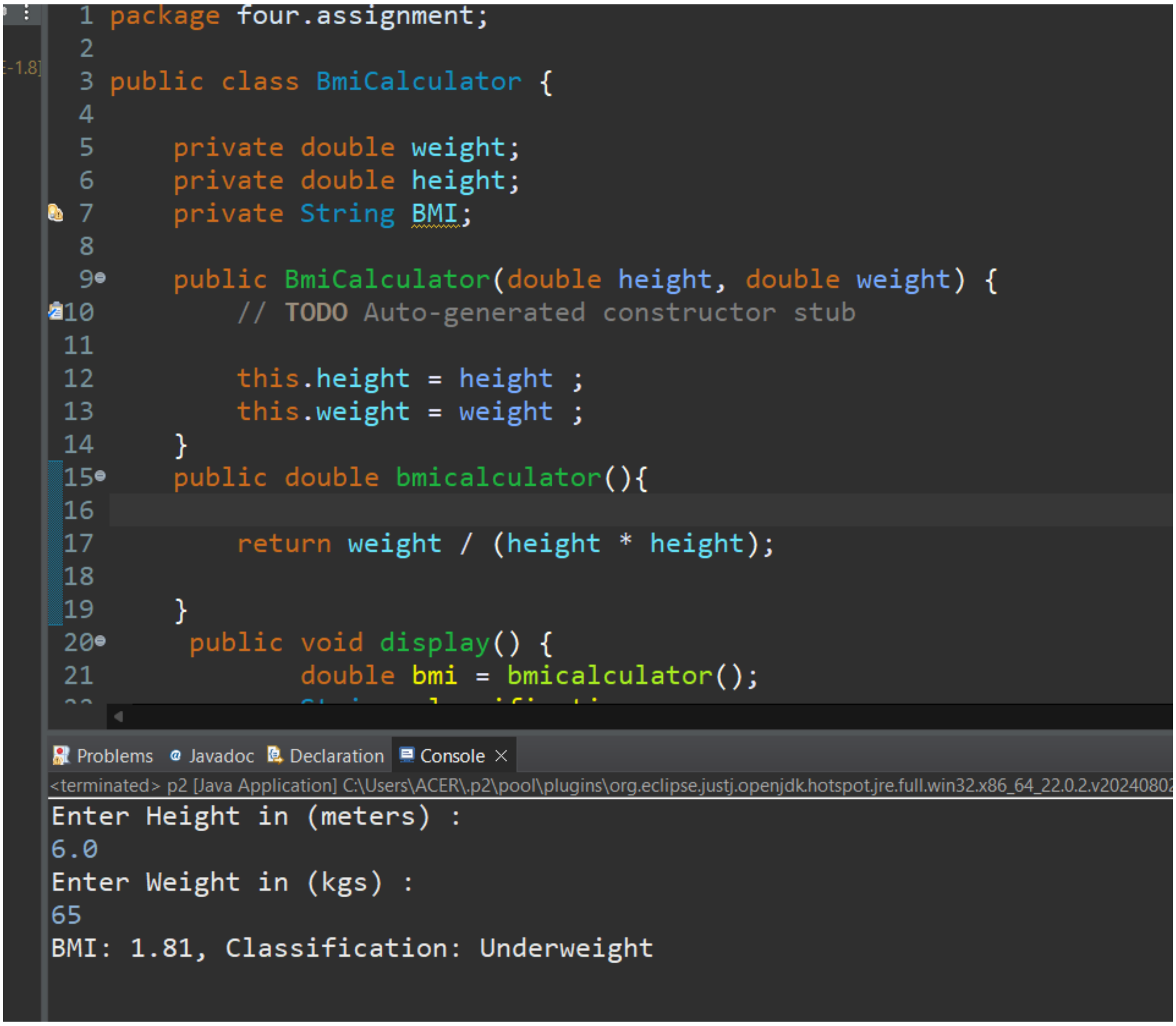
P2.java

```java
package four.assignment;

import java.util.Scanner;

public class p2 {
	public static void main(String []a) {
		Scanner sc = new Scanner(System.in);
		
		System.out.println("Enter Height in (meters) : ");
		double height = sc.nextDouble();
		
		System.out.println("Enter Weight in (kgs) : ");
		double weight = sc.nextDouble()  ;
		
		BmiCalculator calculator = new BmiCalculator(height , weight);
		calculator.display();
	}
}
```

BmiCalculator.java

```java
package four.assignment;

public class BmiCalculator {

	private double weight;
	private double height;
	private String BMI;
	
	public BmiCalculator(double height, double weight) {
		// TODO Auto-generated constructor stub
		
		this.height = height ;
		this.weight = weight ;
	}
	public double bmicalculator(){
		
		return weight / (height * height);
		
	}
	 public void display() {
	        double bmi = bmicalculator();
	        String classification;
	        if (bmi < 18.5) {
	            classification = "Underweight";
	        } else if (bmi < 24.9) {
	            classification = "Normal weight";
```

```
                } else if (bmi < 29.9) {
                    classification = "Overweight";
                } else {
                    classification = "Obese";
                }
                System.out.printf("BMI: %.2f, Classification: %s%n", bmi,
classification);
            }

}
```



```
package four.assignment;

public class BmiCalculator {

    private double weight;
    private double height;
    private String BMI;

    public BmiCalculator(double height, double weight) {
        // TODO Auto-generated constructor stub

        this.height = height ;
        this.weight = weight ;
    }
    public double bmicalculator(){

        return weight / (height * height);

    }
     public void display() {
            double bmi = bmicalculator();
```

```
<terminated> p2 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v2024080
Enter Height in (meters) :
6.0
Enter Weight in (kgs) :
65
BMI: 1.81, 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 (₹).

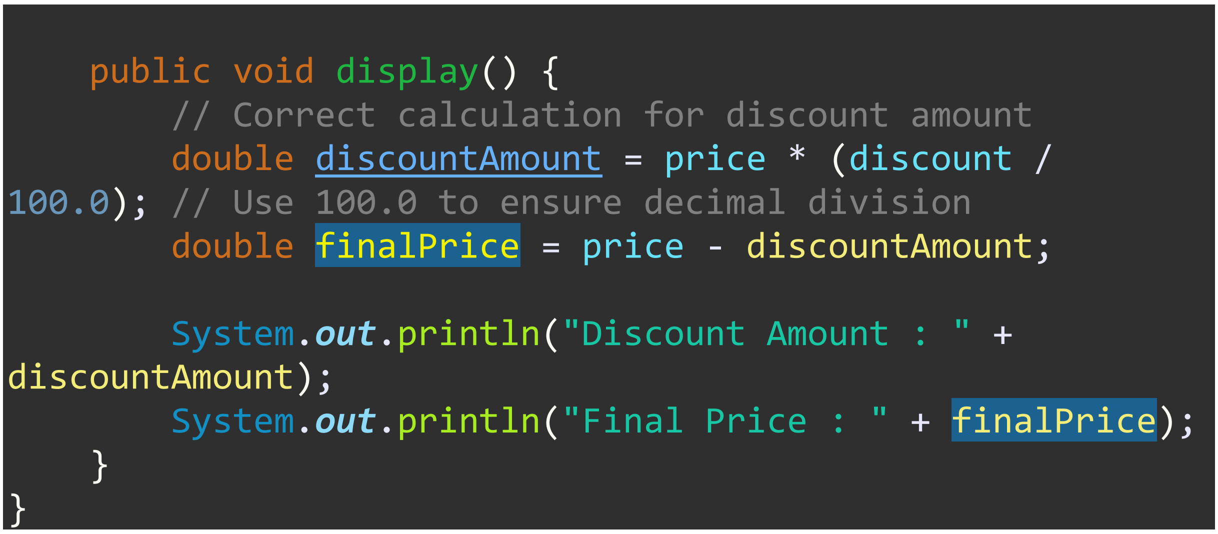
Define the class `DiscountCalculator` with fields, an appropriate constructor, getter and setter methods, a `toString` method, and business logic methods. Define the class `DiscountCalculatorUtil` with methods `acceptRecord`, `printRecord`, and `menuList`. Define the class `Program` with a `main` method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

```java
package four.assignment;

import java.util.Scanner;

public class p4 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the original Price of
Product : ");
        int price = sc.nextInt()    ;

        System.out.println("Enter the Percentage of
Discount :    ");
        int discount = sc.nextInt() ;

        DiscountC calculator = new DiscountC(price ,
discount ) ;
        calculator.display();
    }
}
```

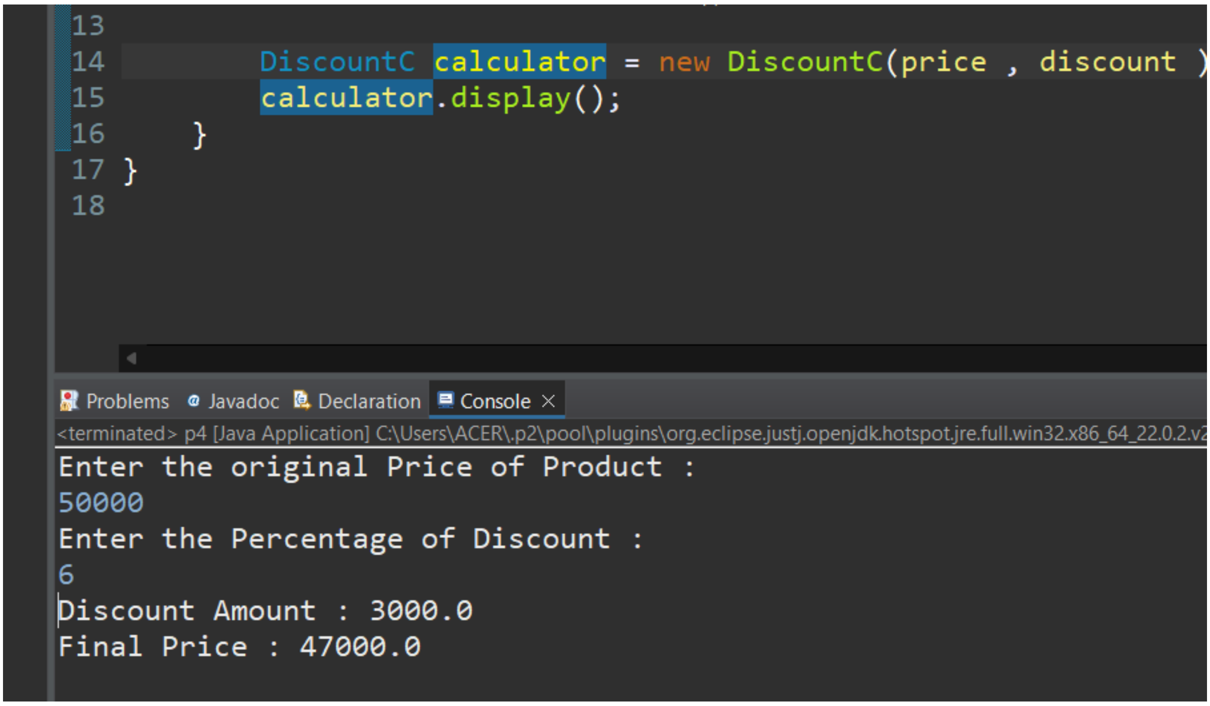
DiscountC.java

```java
package four.assignment;

public class DiscountC {

    private int price;
    private int discount;

    public DiscountC(int price, int discount) {
        // Initializing price and discount
        this.price = price;
        this.discount = discount;
    }
```

ASSIGNMENT NO.- 3



```java
    public void display() {
        // Correct calculation for discount amount
        double discountAmount = price * (discount / 
100.0); // Use 100.0 to ensure decimal division
        double finalPrice = price - discountAmount;

        System.out.println("Discount Amount : " + 
discountAmount);
        System.out.println("Final Price : " + finalPrice);
    }
}
```



```java
            DiscountC calculator = new DiscountC(price , discount )
            calculator.display();
        }
}
```

```
<terminated> p4 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v2
Enter the original Price of Product :
50000
Enter the Percentage of Discount :
6
Discount Amount : 3000.0
Final Price : 47000.0
```

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 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.

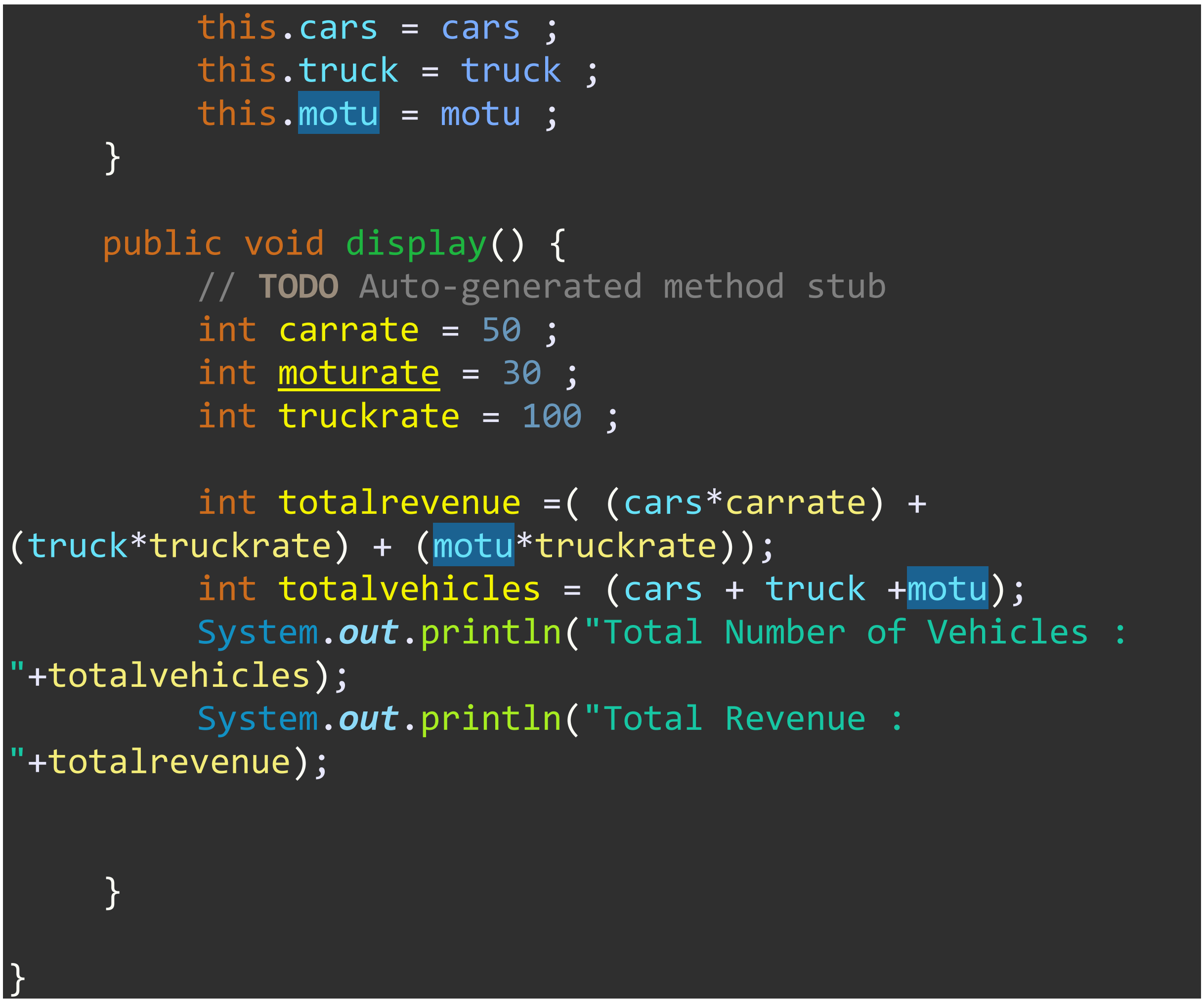
P5.java

```
package four.assignment;

import java.util.Scanner;

public class p5 {
	public static void main(String a[]) {
		Scanner sc = new Scanner(System.in);
		
		System.out.println("Enter number of Cars : ");
		int cars = sc.nextInt() ;
		
		System.out.println("Enter the number of Truck : 
");
		int truck = sc.nextInt()    ;
		
		System.out.println("Enter the number of 
Motorcycle : ");
		int motu = sc.nextInt() ;
		
		tollb manage = new tollb(cars , truck , motu);
		manage.display();
	}
}
```

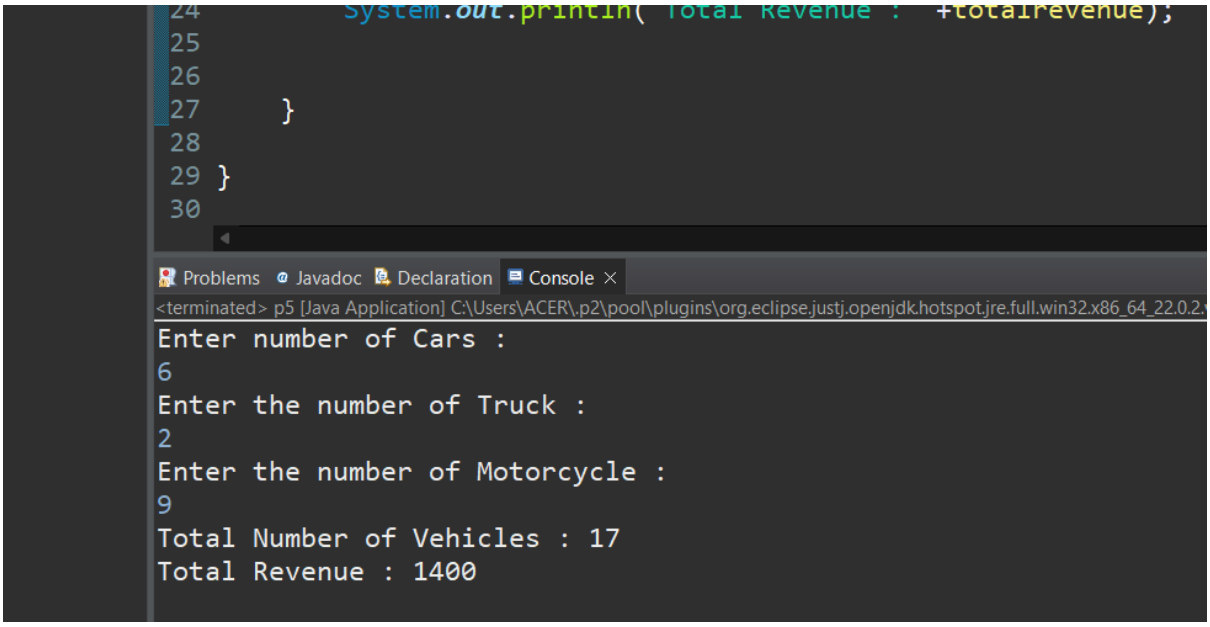
Tollb.java

```
package four.assignment;

public class tollb {
	private int cars;
	private int motu ;
	private int truck ;
	
	public tollb(int cars, int truck, int motu) {
		// TODO Auto-generated constructor stub
```

ASSIGNMENT NO.- 3



```
        this.cars = cars ;
        this.truck = truck ;
        this.motu = motu ;
    }

    public void display() {
        // TODO Auto-generated method stub
        int carrate = 50 ;
        int moturate = 30 ;
        int truckrate = 100 ;

        int totalrevenue =( (cars*carrate) +
(truck*truckrate) + (motu*truckrate));
        int totalvehicles = (cars + truck +motu);
        System.out.println("Total Number of Vehicles :
"+totalvehicles);
        System.out.println("Total Revenue :
"+totalrevenue);


    }

}
```



```
            System.out.println("Total Revenue : "+totalrevenue);


    }

}

```

```
<terminated> p5 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.
Enter number of Cars :
6
Enter the number of Truck :
2
Enter the number of Motorcycle :
9
Total Number of Vehicles : 17
Total Revenue : 1400
```