

Que1)

===== LoanAmortizationCalculator=====

```
public class LoanAmortizationCalculator {  
    private double principal;  
    private double annualInterestRate;  
    private int loanTerm;  
  
    // Constructor  
    public LoanAmortizationCalculator(double principal, double annualInterestRate, int loanTerm) {  
        this.principal = principal;  
        this.annualInterestRate = annualInterestRate;  
        this.loanTerm = loanTerm;  
    }  
  
    // Getters and Setters  
    public double getPrincipal() {  
        return principal;  
    }  
  
    public void setPrincipal(double principal) {  
        this.principal = principal;  
    }  
  
    public double getAnnualInterestRate() {  
        return annualInterestRate;  
    }  
  
    public void setAnnualInterestRate(double annualInterestRate) {  
        this.annualInterestRate = annualInterestRate;  
    }  
}
```

```

public int getLoanTerm() {
    return loanTerm;
}

public void setLoanTerm(int loanTerm) {
    this.loanTerm = loanTerm;
}

// Method to calculate the monthly payment
public double calculateMonthlyPayment() {
    double monthlyInterestRate = annualInterestRate / 12 / 100;
    int numberOfMonths = loanTerm * 12;
    return principal * (monthlyInterestRate * Math.pow(1 + monthlyInterestRate,
numberOfMonths))
        / (Math.pow(1 + monthlyInterestRate, numberOfMonths) - 1);
}

// Method to calculate the total amount paid over the life of the loan
public double calculateTotalAmountPaid() {
    return calculateMonthlyPayment() * loanTerm * 12;
}

// toString method
@Override
public String toString() {
    return String.format("Principal Amount: ₹%.2f\nAnnual Interest Rate: %.2f%%\nLoan Term: %d
years",
        principal, annualInterestRate, loanTerm);
}
}

```

```
=====LoanAmortizationCalculatorUtil.java=====
=====
```

```
import java.util.Scanner;
```

```
public class LoanAmortizationCalculatorUtil {
```

```
    public static LoanAmortizationCalculator acceptRecord() {
```

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

```
        System.out.print("Enter the principal amount of the loan in INR: ");
```

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

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

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

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

```
        int loanTerm = scanner.nextInt();
```

```
        return new LoanAmortizationCalculator(principal, annualInterestRate, loanTerm);
```

```
    }
```

```
    public static void printRecord(LoanAmortizationCalculator loanCalculator) {
```

```
        System.out.println(loanCalculator.toString());
```

```
        System.out.printf("Monthly Payment: ₹%.2f\n", loanCalculator.calculateMonthlyPayment());
```

```
        System.out.printf("Total Amount Paid: ₹%.2f\n", loanCalculator.calculateTotalAmountPaid());
```

```
    }
```

```

public static void menuList() {
    System.out.println("Menu:");
    System.out.println("1. Enter Loan Details");
    System.out.println("2. Display Loan Payment Information");
    System.out.println("3. Exit");
}
}

```

=====Program.java=====

```

import java.util.Scanner;

```

```

public class Program {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        LoanAmortizationCalculator loanCalculator = null;
        int choice;

        do {
            LoanAmortizationCalculatorUtil.menuList();
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    loanCalculator = LoanAmortizationCalculatorUtil.acceptRecord();
                    break;

                case 2:

```

```
        if (loanCalculator != null) {  
            LoanAmortizationCalculatorUtil.printRecord(loanCalculator);  
        } else {  
            System.out.println("Please enter loan details first.");  
        }  
        break;  
  
    case 3:  
        System.out.println("Exiting...");  
        break;  
  
    default:  
        System.out.println("Invalid choice. Please try again.");  
    }  
  
} while (choice != 3);  
  
scanner.close();  
}  
}
```

Que 2)

```
=====
CompoundInterestCalculator.java=====
=====
```

```
public class CompoundInterestCalculator {

    private double principal;

    private double annualInterestRate;

    private int numberOfCompounds;

    private int years;


    // Constructor

    public CompoundInterestCalculator(double principal, double annualInterestRate, int
numberOfCompounds, int years) {

        this.principal = principal;

        this.annualInterestRate = annualInterestRate;

        this.numberOfCompounds = numberOfCompounds;

        this.years = years;

    }


    // Getters and Setters

    public double getPrincipal() {

        return principal;

    }


    public void setPrincipal(double principal) {

        this.principal = principal;

    }


    public double getAnnualInterestRate() {

        return annualInterestRate;

    }

}
```

```
}
```

```
public void setAnnualInterestRate(double annualInterestRate) {  
    this.annualInterestRate = annualInterestRate;  
}
```

```
public int getNumberOfCompounds() {  
    return numberOfCompounds;  
}
```

```
public void setNumberOfCompounds(int numberOfCompounds) {  
    this.numberOfCompounds = numberOfCompounds;  
}
```

```
public int getYears() {  
    return years;  
}
```

```
public void setYears(int years) {  
    this.years = years;  
}
```

```
// Method to calculate the future value of the investment  
public double calculateFutureValue() {  
    return principal * Math.pow(1 + annualInterestRate / numberOfCompounds,  
numberOfCompounds * years);  
}
```

```
// Method to calculate the total interest earned  
public double calculateTotalInterest() {  
    return calculateFutureValue() - principal;  
}
```

```

    }

    // toString method
    @Override
    public String toString() {
        return String.format("Principal Amount: ₹%.2f\nAnnual Interest Rate: %.2f%%\nNumber of
Compounds per Year: %d\nInvestment Duration: %d years",
            principal, annualInterestRate, numberOfCompounds, years);
    }
}

```

```

=====
CompoundInterestCalculatorUtil.java=====

```

```

import java.util.Scanner;

public class CompoundInterestCalculatorUtil {

    public static CompoundInterestCalculator acceptRecord() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the initial investment amount in INR: ");
        double principal = scanner.nextDouble();

        System.out.print("Enter the annual interest rate (in percentage): ");
        double annualInterestRate = scanner.nextDouble();

        System.out.print("Enter the number of times interest is compounded per year: ");
        int numberOfCompounds = scanner.nextInt();

        System.out.print("Enter the investment duration (in years): ");
        int years = scanner.nextInt();

        return new CompoundInterestCalculator(principal, annualInterestRate / 100,
            numberOfCompounds, years);
    }
}

```



```

public static void printRecord(CompoundInterestCalculator calculator) {
    System.out.println(calculator.toString());
    System.out.printf("Future Value: ₹%.2f\n", calculator.calculateFutureValue());
    System.out.printf("Total Interest Earned: ₹%.2f\n", calculator.calculateTotalInterest());
}

```

```

public static void menuList() {
    System.out.println("Menu:");
    System.out.println("1. Enter Investment Details");
    System.out.println("2. Display Investment Information");
    System.out.println("3. Exit");
}

```

```

}

```

```

=====program.java=====

```

```

import java.util.Scanner;

```

```

public class Program {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        CompoundInterestCalculator calculator = null;
        int choice;

        do {
            CompoundInterestCalculatorUtil.menuList();
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    calculator = CompoundInterestCalculatorUtil.acceptRecord();

```

```

        break;

    case 2:
        if (calculator != null) {
            CompoundInterestCalculatorUtil.printRecord(calculator);
        } else {
            System.out.println("Please enter investment details first.");
        }
        break;

    case 3:
        System.out.println("Exiting...");
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
    }

} while (choice != 3);

scanner.close();
}
}

```

Que 3)

=====BMITracker.java=====

```

public class BMITracker {
    private double weight; // in kilograms
    private double 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 BMI  
public String classifyBMI() {
```

```

double bmi = calculateBMI();
if (bmi < 18.5) {
    return "Underweight";
} else if (bmi < 24.9) {
    return "Normal weight";
} else if (bmi < 29.9) {
    return "Overweight";
} else {
    return "Obese";
}
}

```

```
// toString method
```

```
@Override
```

```
public String toString() {
```

```
    return String.format("Weight: %.2f kg\nHeight: %.2f meters\nBMI: %.2f\nClassification: %s",
        weight, height, calculateBMI(), classifyBMI());
}
```

```
}
```

```
=====BMITrackerUtil.java=====
```

```
import java.util.Scanner;
```

```
public class BMITrackerUtil {
```

```
    public static BMITracker acceptRecord() {
```

```
        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();

        return new BMITracker(weight, height);
    }

    public static void printRecord(BMITracker bmiTracker) {
        System.out.println(bmiTracker.toString());
    }

    public static void menuList() {
        System.out.println("Menu:");
        System.out.println("1. Enter BMI Details");
        System.out.println("2. Display BMI Information");
        System.out.println("3. Exit");
    }
}

=====Program.java=====

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        BMITracker bmiTracker = null;
        int choice;

        do {
            BMITrackerUtil.menuList();
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();

```

```

switch (choice) {
    case 1:
        bmiTracker = BMITrackerUtil.acceptRecord();
        break;

    case 2:
        if (bmiTracker != null) {
            BMITrackerUtil.printRecord(bmiTracker);
        } else {
            System.out.println("Please enter BMI details first.");
        }
        break;

    case 3:
        System.out.println("Exiting...");
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
}

} while (choice != 3);

scanner.close();
}
}

```

---

Que4

DiscountCalculator.java=====

```

public class DiscountCalculator {

```

```
private double originalPrice;

private double discountRate;


// Constructor

public DiscountCalculator(double originalPrice, double discountRate) {

    this.originalPrice = originalPrice;

    this.discountRate = discountRate;

}


// Getters and Setters

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;

}


// Method to calculate the discount amount

public double calculateDiscountAmount() {

    return originalPrice * (discountRate / 100);

}
```

```

// Method to calculate the final price after discount
public double calculateFinalPrice() {
    return originalPrice - calculateDiscountAmount();
}

// toString method
@Override
public String toString() {
    return String.format("Original Price: ₹%.2f\nDiscount Rate: %.2f%%\nDiscount Amount: ₹%.2f\nFinal Price: ₹%.2f",
        originalPrice, discountRate, calculateDiscountAmount(), calculateFinalPrice());
}
}

```

DiscountCalculatorUtil.java=====

```
import java.util.Scanner;
```

```

public class DiscountCalculatorUtil {

    public static DiscountCalculator acceptRecord() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the original price of the item in INR: ");
        double originalPrice = scanner.nextDouble();
        System.out.print("Enter the discount percentage: ");
        double discountRate = scanner.nextDouble();

        return new DiscountCalculator(originalPrice, discountRate);
    }
}

```



```
public static void printRecord(DiscountCalculator discountCalculator) {  
    System.out.println(discountCalculator.toString());  
}
```

```
public static void menuList() {  
    System.out.println("Menu:");  
    System.out.println("1. Enter Item Details");  
    System.out.println("2. Display Discount Information");  
    System.out.println("3. Exit");  
}  
}
```

Program.java

```
import java.util.Scanner;
```

```
public class Program {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        DiscountCalculator discountCalculator = null;  
        int choice;  
  
        do {  
            DiscountCalculatorUtil.menuList();  
            System.out.print("Choose an option: ");  
            choice = scanner.nextInt();  
  
            switch (choice) {  
                case 1:  
                    discountCalculator = DiscountCalculatorUtil.acceptRecord();  
                    break;
```

case 2:

```
    if (discountCalculator != null) {  
        DiscountCalculatorUtil.printRecord(discountCalculator);  
    } else {  
        System.out.println("Please enter item details first.");  
    }  
    break;
```

case 3:

```
    System.out.println("Exiting...");  
    break;
```

default:

```
    System.out.println("Invalid choice. Please try again.");  
}
```

```
} while (choice != 3);
```

```
    scanner.close();
```

```
}
```

```
}
```

Que 5)

TollBoothRevenueManager.java

```
public class TollBoothRevenueManager {  
    private double carTollRate;  
    private double truckTollRate;  
    private double motorcycleTollRate;
```

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

```
// Constructor
```

```
public TollBoothRevenueManager(double carTollRate, double truckTollRate, double  
motorcycleTollRate) {
```

```
    this.carTollRate = carTollRate;  
    this.truckTollRate = truckTollRate;  
    this.motorcycleTollRate = motorcycleTollRate;  
    this.numberOfCars = 0;  
    this.numberOfTrucks = 0;  
    this.numberOfMotorcycles = 0;
```

```
}
```

```
// Getters and Setters
```

```
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;  
}
```

```
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 (numberOfCars * carTollRate) + (numberOfTrucks * truckTollRate) +
        (numberOfMotorcycles * motorcycleTollRate);
    }

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

    // toString method
    @Override
    public String toString() {
        return String.format("Car Toll Rate: ₹%.2f\nTruck Toll Rate: ₹%.2f\nMotorcycle Toll Rate:
        ₹%.2f\n" +
            "Number of Cars: %d\nNumber of Trucks: %d\nNumber of Motorcycles: %d\n" +
            "Total Vehicles: %d\nTotal Revenue: ₹%.2f",
            carTollRate, truckTollRate, motorcycleTollRate, numberOfCars, numberOfTrucks,
            numberOfMotorcycles,
            calculateTotalVehicles(), calculateTotalRevenue());
    }
}

TollBoothRevenueManagerUtil=====

import java.util.Scanner;

public class TollBoothRevenueManagerUtil {

    public static TollBoothRevenueManager acceptRecord() {

```

```

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the toll rate for cars in INR: ");
double carTollRate = scanner.nextDouble();
System.out.print("Enter the toll rate for trucks in INR: ");
double truckTollRate = scanner.nextDouble();
System.out.print("Enter the toll rate for motorcycles in INR: ");
double motorcycleTollRate = scanner.nextDouble();

TollBoothRevenueManager tollBooth = new TollBoothRevenueManager(carTollRate,
truckTollRate, motorcycleTollRate);

System.out.print("Enter the number of cars passing through: ");
tollBooth.setNumberOfCars(scanner.nextInt());
System.out.print("Enter the number of trucks passing through: ");
tollBooth.setNumberOfTrucks(scanner.nextInt());
System.out.print("Enter the number of motorcycles passing through: ");
tollBooth.setNumberOfMotorcycles(scanner.nextInt());

return tollBooth;
}

public static void printRecord(TollBoothRevenueManager tollBooth) {
    System.out.println(tollBooth.toString());
}

public static void menuList() {
    System.out.println("Menu:");
    System.out.println("1. Enter Toll Rates and Vehicle Counts");
    System.out.println("2. Display Toll Booth Information");
    System.out.println("3. Exit");
}

```

```
}  
}
```

Program.java

```
import java.util.Scanner;
```

```
public class Program {
```

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

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

```
        TollBoothRevenueManager tollBooth = null;
```

```
        int choice;
```

```
        do {
```

```
            TollBoothRevenueManagerUtil.menuList();
```

```
            System.out.print("Choose an option: ");
```

```
            choice = scanner.nextInt();
```

```
            switch (choice) {
```

```
                case 1:
```

```
                    tollBooth = TollBoothRevenueManagerUtil.acceptRecord();
```

```
                    break;
```

```
                case 2:
```

```
                    if (tollBooth != null) {
```

```
                        TollBoothRevenueManagerUtil.printRecord(tollBooth);
```

```
                    } else {
```

```
                        System.out.println("Please enter toll rates and vehicle counts first.");
```

```
                    }
```

```
                    break;
```

```
                case 3:
```

```
                    System.out.println("Exiting...");
```

```
break;
```

```
default:
```

```
    System.out.println("Invalid choice. Please try again.");
```

```
}
```

```
} while (choice != 3);
```

```
scanner.close();
```

```
}
```

```
}
```