

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 (₹).

Define class `LoanAmortizationCalculator` with methods `acceptRecord`, `calculateMonthlyPayment` & `printRecord` and test the functionality in `main` method.

```
package pack1;
```

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

```
class LoanAmortizationCalculator {  
    double principalamount;    float  
    annualinterestrate;    int loanterm;
```

```
    public void acceptRecord() {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter Principal Amount: ");  
        principalamount = sc.nextDouble();  
  
        System.out.println("Enter Annual Interest Rate: ");  
        annualinterestrate = sc.nextFloat();  
  
        System.out.println("Enter the Loan Term (in years): ");  
        loanterm = sc.nextInt();  
        sc.close();  
    }  
}
```

```

    public double calculateMonthlyPayment() {
        double monthlyInterestRate = (annualInterestRate / 12) / 100;
        int numberOfMonths = loanTerm * 12;    double
        monthlyPayment = principalAmount *
            (monthlyInterestRate * Math.pow(1 + monthlyInterestRate, numberOfMonths)) /
            (Math.pow(1 + monthlyInterestRate, numberOfMonths) - 1);

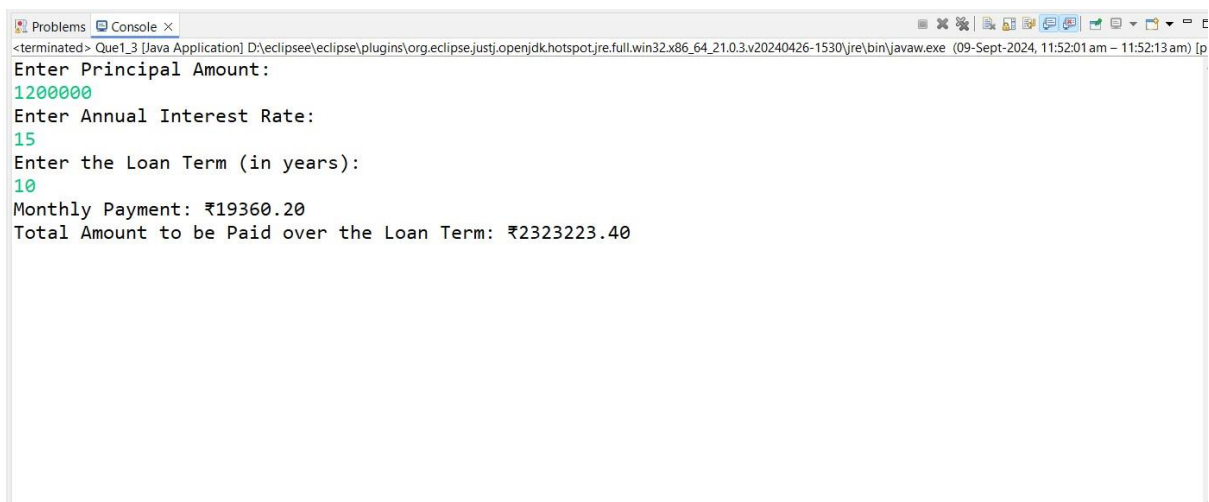
        return monthlyPayment;
    }

    public void printRecord() {
        double monthlyPayment = calculateMonthlyPayment();
        double totalAmountPaid = monthlyPayment * loanTerm * 12;

        System.out.printf("Monthly Payment: ₹%.2f\n", monthlyPayment);
        System.out.printf("Total Amount to be Paid over the Loan Term: ₹%.2f\n", totalAmountPaid);
    }
}

public class Que1_3 {
    public static void main(String[] args) {
        LoanAmortizationCalculator l = new LoanAmortizationCalculator();
        l.acceptRecord();
        l.printRecord();
    }
}

```



```

Enter Principal Amount:
1200000
Enter Annual Interest Rate:
15
Enter the Loan Term (in years):
10
Monthly Payment: ₹19360.20
Total Amount to be Paid over the Loan Term: ₹2323223.40

```

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:**

$$\square \text{ futureValue} = \text{principal} * (1 + \text{annualInterestRate} / \text{numberOfCompounds})^{(\text{numberOfCompounds} * \text{years})}$$

- **Total Interest Earned:** $\text{totalInterest} = \text{futureValue} - \text{principal}$

3. Display the future value and the total interest earned, in Indian Rupees (₹).

Define class CompoundInterestCalculator with methods acceptRecord , calculateFutureValue, printRecord and test the functionality in main method.

```
package pack_comp; import
java.util.Scanner; class
CompoundInterestCalculator{
float principal; float
annualInterestRate; int
noOfTimes;
int duration;

public void acceptRecord() {
Scanner sc=new Scanner(System.in);
System.out.println("Enter initial ammount:");
principal=sc.nextFloat();

System.out.println("Enter annual interest rate:");
annualInterestRate=sc.nextFloat();

System.out.println("Enter the no. of times interest compounded in year");
noOfTimes=sc.nextInt();
System.out.println("Enter the duration:");
duration=sc.nextInt();
sc.close();
}

public double calculteFutureRecord() {
double ratePerCompound = annualInterestRate / noOfTimes / 100;
int totalCompounds = noOfTimes * duration;
```

```

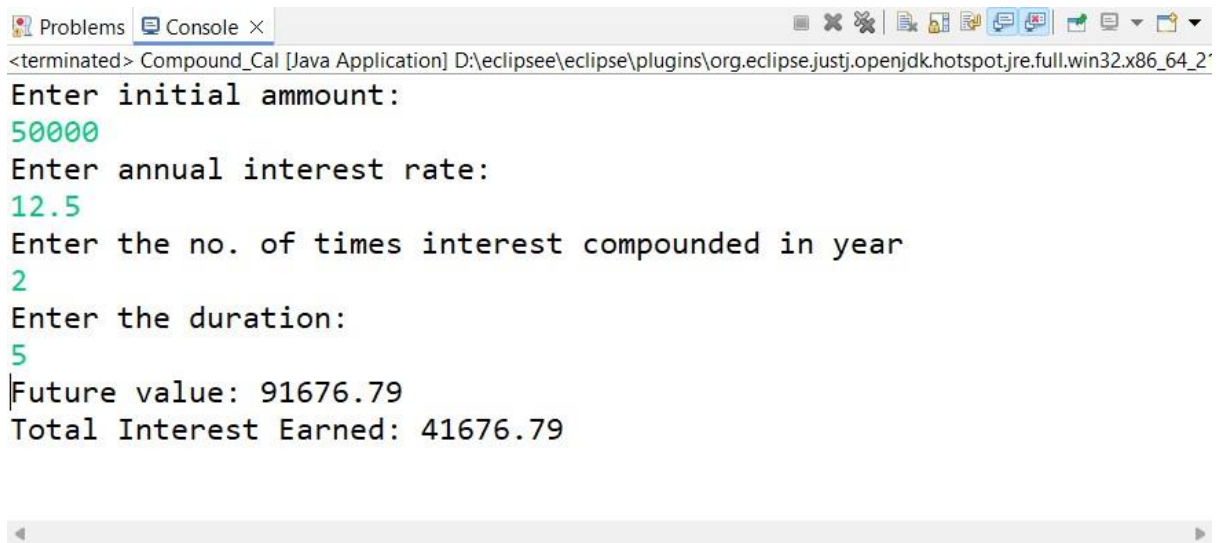
        double futureValue = principal * Math.pow(1 + ratePerCompound, totalCompounds);
        return futureValue;
    }

    public void printRecord() {
double x=calculteFutureRecord();
double totalInterest=x-principal;
        System.out.printf("Future value: %.2f \n",x);
        System.out.printf("Total Interest Earned: %.2f \n",totalInterest);
    }

}

public class Compound_Cal {
    public static void main(String[] args) {
        CoumpoundInterestCalculator c= new CoumpoundInterestCalculator();
        c.acceptRecord();
        c.printRecord();
    }
}

```



```

<terminated> Compound_Cal [Java Application] D:\eclipse\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
Enter initial ammount:
50000
Enter annual interest rate:
12.5
Enter the no. of times interest compounded in year
2
Enter the duration:
5
Future value: 91676.79
Total Interest Earned: 41676.79

```

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

Define class BMITracker with methods acceptRecord, calculateBMI, classifyBMI & printRecord and test the functionality in main method.

```
package pack_bmi;
```

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

```
class BMITracker{
    float weight;
    float height;

    public void acceptRecord() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the weight: ");
        weight = sc.nextFloat();

        System.out.println("Enter the height: ");
        height = sc.nextFloat();
    }
    sc.close();

    public float BmiCalc() {
        float bmi=(weight/(height*height));
    }
    return bmi;

    public void classifyBmi() {
        float x=BmiCalc();
        System.out.println("Your BMI is:"+x);
        if(x<18.5) {
            System.out.println("Underweight...");
        }
        else if(x>=18.5 && x<=24.9) {
            System.out.println("Normal weight...");
        }
    }
}
```

```

        else if(x>=25 && x< 29.9) {
            System.out.println("Overweight");
        }
        else {
            System.out.println("Obese...");
        }
    }
}

public class BMICalc {
    public static void main(String[] args) {
        BMITracker n= new BMITracker();
        n.acceptRecord();
        //      n.BmiCalc();
        n.classifyBmi();}
}

```

```

<terminated> BMICalc [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20
Enter the weight:
90
Enter the height:
2.1
Your BMI is:20.408167
Normal weight...

```

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 (₹). Define class DiscountCalculator with methods acceptRecord, calculateDiscount & printRecord and test the functionality in main method.

```
package pack_eg;
```

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

```

class DiscCal {    float
original_price;
    float discount_rate;

    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the original price: ");    original_price
        = sc.nextFloat();

        System.out.print("Enter the discount percentage: ");
        discount_rate = sc.nextFloat();

        sc.close();
    }

    public float[] calculate() {
        float discountAmount = original_price * (discount_rate / 100);
        float finalPrice = original_price - discountAmount;

        return new float[]{discountAmount, finalPrice};
    }

    public void display() {    float[]
result = calculate();    float
discountAmount = result[0];
    float finalPrice = result[1];

    System.out.printf("The discount amount is: ₹%.2f\n", discountAmount);
    System.out.printf("The final price after discount is: ₹%.2f\n", finalPrice);
    }
}

public class DiscountCalculator {
    public static void main(String[] args) {
        DiscCal d = new DiscCal();
        d.acceptRecord();
        d.display();
    }
}

```

```
Problems Console x
<terminated> DiscountCalculator [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (09-Sept-2024, 12:01:00 pm - 12:01:
Enter the original price: 2000
Enter the discount percentage: 25
The discount amount is: ₹500.00
The final price after discount is: ₹1500.00
```

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 class TollBoothRevenueManager with methods

acceptRecord, setTollRates, calculateRevenue & printRecord and test the functionality in main method.

```
package pack_toll;
```

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

```
class Toll{
```

```
    private int carRate;
```

```
    private int truckRate;
```

```
    private int bikeRate;
```

```
    private int carCount;
```

```
    private int truckCount;
```

```
    private int bikeCount;
```

```
    //public Toll() {}
```

```
    public Toll(int carRate, int truckRate, int bikeRate, int carCount, int truckCount, int
```

```
    bikeCount) {
```

```
        this.carRate = carRate;
```

```
        this.truckRate = truckRate;
```

```
        this.bikeRate = bikeRate;
```

```
        this.carCount = carCount;
```

```
        this.truckCount = truckCount;
```

```
        this.bikeCount = bikeCount;
```

```
    }
```



```

private static Scanner sc=new Scanner(System.in);

public void acceptRates(){
    System.out.print("Enter the Car Rate :");
    carRate=sc.nextInt();
    System.out.print("Enter the Truck Rate :");
    truckRate=sc.nextInt();
    System.out.print("Enter the Bike Rate :");
    bikeRate=sc.nextInt();

}

public void setCount() {
    System.out.print("Enter the Car Count :");
    carCount=sc.nextInt();
    System.out.print("Enter the Truck Count :");
    truckCount=sc.nextInt();
    System.out.print("Enter the Bike Count :");
    bikeCount=sc.nextInt();

}

public void calculateToll() {
    int x=(carRate*carCount);
    int y=(truckRate*truckCount);
    int z=(bikeRate*bikeCount);
    int totalRevenue = (carRate*carCount) + (truckRate*truckCount) +
(bikeRate*truckCount);
    int totalVehicles = carCount + truckCount + bikeCount;
    System.out.println("Toll Rate of Car: "+x);
    System.out.println("Toll Rate of Truck: "+y);
    System.out.println("Toll Rate of Bike: "+z);
    System.out.println("Total Revenue Generated : "+totalRevenue);
    System.out.println("Total Vehicles Passed : "+totalVehicles);

    sc.close();
}
}

public class MainToll {
public static void main(String[] args) {
    Toll t= new Toll(0,0,0,0,0,0);

    t.acceptRates();
    t.setCount();
    t.calculateToll();}

}

```

```
<terminated> MainToll [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (09-Sept-2024, 12:46:59 pm - 12:47:25 pm) [pi
Enter the Car Rate :500
Enter the Truck Rate :1000
Enter the Bike Rate :250
Enter the Car Count :50
Enter the Truck Count :100
Enter the Bike Count :150
Toll Rate of Car: 25000
Toll Rate of Truck: 100000
Toll Rate of Bike: 37500
Total Revenue Generated : 150000
Total Vehicles Passed : 300
```