

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 org.programming;

import java.util.Scanner;

class LoanAmortizationCalculator{

    float principal;

    float rate;

    int years;

    double monthlyPayment;

    int n;

    Scanner sc=new Scanner(System.in);

    public void acceptRecord() {

        System.out.println("Enter Principal:  ");

        principal=sc.nextFloat();

        System.out.println("Enter rate:      ");

        rate=sc.nextFloat();

        System.out.println("Enter years:    ");

        years=sc.nextInt();

    }

    public void calculateMonthlyPayment() {

        double monthlyInterestRate= rate/12/100;

        n=years*12;

        monthlyPayment =

principal*(monthlyInterestRate*(Math.pow((1+monthlyInterestRate),n)))/(Math

.pow((1+monthlyInterestRate),n)-1);

    }

    public void printRecord() {

        System.out.println("Monthly Payment:"+ monthlyPayment);
    }
}

```

```

        System.out.println("Total Amount Paid
:"+monthlyPayment*n);
    }
}

```

```

public class programming{
    public static void main(String[] args) {

        LoanAmortizationCalculator l = new
LoanAmortizationCalculator();
        l.acceptRecord();
        l.calculateMonthlyPayment();
        l.printRecord();
    }
}

```

```

Enter Principal:
1000000
Enter rate:
8
Enter years:
25
Monthly Payment :7718.162322056645
Total Amount Paid :2315448.6966169938

```

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:**
 - $\text{futureValue} = \text{principal} * (1 + \text{annualInterestRate} / \text{numberOfCompounds})^{(\text{numberOfCompounds} * \text{years})}$
 - **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 org.programming2;

import java.util.Scanner;

class CompoundInterestCalculator {
    private double principal;
    private double annualInterest;
    private int noOfCompound;
    private int years;
    private double futureValue;
    private double totalInterest;
    Scanner sc=new Scanner(System.in);
    public void acceptRecord() {
        System.out.println("Initial Investment Ammount:");
```

```

        principal=sc.nextDouble();
        System.out.println("Annual Interest Rate:");
        annualInterest=sc.nextDouble();
        System.out.println("Number of time interest compounded:");
        noOfCompound=sc.nextInt();
        System.out.println("Enter year:");
        years=sc.nextInt();
    }

    public void calculateFutureValue() {
        futureValue= principal *Math.pow ((1 + annualInterest /
noOfCompound),noOfCompound * years);
        totalInterest=futureValue-principal;

    }

    public void printRecord() {
        System.out.println("Future Value of investment:
"+futureValue);
        System.out.println("Total Interest Earned: "+totalInterest);
    }

}

public class programming {
    public static void main(String[] args) {
        CompoundInterestCalculator c1=new
CompoundInterestCalculator();
        c1.acceptRecord();
        c1.calculateFutureValue();
        c1.printRecord();
    }
}

```

```
}  
}
```

Initial Investment Ammount:

10000

Annual Interest Rate:

6

Number of time interest compounded:

4

Enter year:

10

Future Value of investment: 8.271806125530277E19|

Total Interest Earned: 8.271806125530276E19

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 org.programming3;
```

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

```
class BmiTracker{  
    private float weight;  
    private float height;  
    private float bmi;  
    Scanner sc=new Scanner(System.in);  
    public void acceptRecord() {  
        System.out.println("Enter weight: ");  
        weight=sc.nextFloat();  
        System.out.println("Enter height");  
        height=sc.nextFloat();  
    }  
}
```

```
}
```

```
public void calculateBMI() {  
    bmi=weight/(weight*height);  
}
```

```
public void classifyBMI() {  
    if(bmi<18.5)  
        System.out.println("Underweight");  
    else if(bmi>=18.5 && bmi<=24.9)  
        System.out.println("Normal weight");  
    else if (bmi>=25 && bmi<=29.9)  
        System.out.println("Overweight");  
    else  
        System.out.println("Obese");  
}
```

```
public void printRecord () {  
    System.out.println("BMI:      "+bmi);  
  
}
```

```
}
```

```
public class programming{  
    public static void main(String args[])  
    {  
        BmiTracker bt=new BmiTracker();  
        bt.acceptRecord();  
    }  
}
```



```
        bt.calculateBMI();  
        bt.printRecord();  
        bt.classifyBMI();  
    }  
  
}
```

```
Enter weight:  
60  
Enter height  
4.11  
BMI:      0.24330899  
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 (₹).

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

```
package org.programming4;
```

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

```
class DiscountCalculator{
```

```
    private float price;
```

```
    private int discount;
```

```
    private float disAmount;
```

```
    private float finalPrice;
```

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

```
    public void acceptRecord() {
```

```
        System.out.println("Enter original price:");
```

```

        price =sc.nextFloat();
        System.out.println("Enter discount:  ");
        discount=sc.nextInt();
    }
    public void calculateDiscount() {
        disAmount= price* discount/100;
        finalPrice=price-disAmount;
    }
    public void printRecord() {
        System.out.println("Discount Amount: "+disAmount);
        System.out.println("Final Price: "+finalPrice);
    }
}

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

```

```

Enter original price:
1400
Enter discount:
10
Discount Amount:  140.0
Final Price:  1260.0

```

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 class TollBoothRevenueManager with methods acceptRecord, setTollRates, calculateRevenue & printRecord and test the functionality in main method.

```
package org.programming5;
```

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

```
class TollBoothRevenueManager {  
    private int carToll;  
    private int truckToll;  
    private int motorcycleToll;  
    private int noCar;  
    private int noTruck;  
    private int noMotorcycle;
```

```

private float totalRevenue;

private int totVehicles;

Scanner sc=new Scanner(System.in);


public void acceptRecord() {
System.out.println("Enter no. of Cars: ");
noCar=sc.nextInt();
System.out.println("Enter no. of Truck: ");
noTruck=sc.nextInt();
System.out.println("Enter no. of Motorcycle: ");
        noMotorcycle=sc.nextInt();

}

public void setTollRates()
{
        System.out.println("Enter toll rate for car:  ");
        carToll=sc.nextInt();
        System.out.println("Enter toll rate for truck: ");
        truckToll=sc.nextInt();
        System.out.println("Enter toll rate for motorcycle: ");
        motorcycleToll=sc.nextInt();
}

public void calculateRevenue() {
        totVehicles=noCar+noTruck+noMotorcycle;

        totalRevenue=(noCar*carToll)+(noTruck*truckToll)+(noMotorcycle*motorcycleToll);
}

```

```

        public void printRecord(){
            System.out.println("Total number of Vehicles:
"+totVehicles);

            System.out.println("Total Revenues: "+totalRevenue);
        }
    }

    public class programming{
        public static void main(String args[])
        {
            TollBoothRevenueManager t=new TollBoothRevenueManager();
            t.setTollRates();
            t.acceptRecord();
            t.calculateRevenue();
            t.printRecord();
        }
    }

```

```

Enter toll rate for car:
50
Enter toll rate for truck:
100
Enter toll rate for motorcycle:
30
Enter no. of Cars:
2

```

```

Enter no. of Truck:
4
Enter no. of Motorcycle:
6
Total number of Vehicles: 12
Total Revenues: 680.0

```