

## Assignment 4

### 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 Assignment_4;
import java.util.Scanner;
class LoanAmortizationCalculator{
    private float principal;
    private float annualInterestRate;
    private int loanTerm;
    private float monthlyPayment;
    private float monthlyInterestRate;
    private int numberOfMonths;
    public float getPrincipal() {
        return principal;
    }
    public void setPrincipal(float principal) {
        this.principal = principal;
    }
    public float getAnnualInterestRate() {
        return annualInterestRate;
    }
    public void setAnnualInterestRate(float annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
    }
    public int getLoanTerm() {
        return loanTerm;
    }
}
```

```

public void setLoanTerm(int loanTerm) {
    this.loanTerm = loanTerm;
}
public void acceptRecord(float principal,float annualInterestRate, int loanTerm) {
    this.principal=principal;
    this.annualInterestRate=annualInterestRate;
    this.loanTerm=loanTerm;
}
public void calculateMonthlyPayment() {
    monthlyInterestRate = (float)annualInterestRate / 12 / 100 ;
    numberOfMonths = (int)loanTerm * 12;
    monthlyPayment = (float) (principal * (monthlyInterestRate * Math.pow((1 +
    monthlyInterestRate),numberOfMonths)) / (Math.pow((1 +
    monthlyInterestRate),numberOfMonths) - 1));
}
public void printRecord() {
    System.out.println("Monthly payment is: "+monthlyPayment);
}
}
public class Q1 {
    public static void main(String[] args) {
        LoanAmortizationCalculator lc=new LoanAmortizationCalculator();
        Scanner sc=new Scanner(System.in);
        lc.setPrincipal(sc.nextFloat());
        lc.setAnnualInterestRate(sc.nextFloat());
        lc.setLoanTerm(sc.nextInt());
        // lc.acceptRecord(sc.nextFloat(),sc.nextFloat(), sc.nextInt());
        //
        lc.calculateMonthlyPayment();
        lc.printRecord();
    }
}

```

Output:

```

<terminated> Q1 (1) [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (09-Sep-2022 12:30:00)
12300
2.4
3
Monthly payment is: 354.46008

```

## 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 Assignment_4;
import java.util.Scanner;
class CompoundInterestCalculator {
    private float principal;
    private float annualInterestRate;
    private int numberOfCompounds;
    private int years;
    private float futureValue;
    private float totalInterest;
    public void setPrincipal(float principal) { // sets
        this.principal = principal;
        System.out.println("New principal is" + principal);
    }
    public float getPrincipal() { // get from user
        return this.principal;
    }
    public void setAnnualInterest(float annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
    }
    public float getAnnualInterest() {
        return this.annualInterestRate;
    }
    public void setNumberOfCompounds(int numberOfCompounds) {
        this.numberOfCompounds = numberOfCompounds;
    }
    public int getNumberOfCompounds() {
        return this.numberOfCompounds;
    }
    public void setYears(int years) {
        this.years = years;
    }
}
```

```

public int getYears() {
    return years;
}
public void acceptRecord() {
    Scanner sc = new Scanner(System.in);
    setPrincipal(sc.nextFloat());
    setAnnualInterest(sc.nextFloat());
    setNumberOfCompounds(sc.nextInt());
    setYears(sc.nextInt());
    sc.close();
}
public void calculateFutureValue() {
    futureValue = (float) (principal
    * Math.pow(1 + (annualInterestRate / numberOfCompounds), numberOfCompounds * years));
    totalInterest = (float) (futureValue - principal);
}
public void printRecord() {
    System.out.println("Future value is:" + futureValue);
    System.out.println("Total interest earned: " + totalInterest);
}
}
public class Q2 {
    public static void main(String[] args) {
        CompoundInterestCalculator ci = new CompoundInterestCalculator();
        ci.acceptRecord();
        ci.calculateFutureValue();
        ci.printRecord();
    }
}

```

Output:

```

C:\Program Files\Java\jdk-1.8.0_101\bin> java Q2 (1) java Application
12300
New principal is12300.0
3.3
3
2
Future value is:1054923.0
Total interest earned: 1042623.0

```

### 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 Assignment_4;
import java.util.Scanner;
class BMITracker {
    private float weight;
    private float height;
    private float bmiCalculate;
    public float getWeight() {
        return weight;
    }
    public void setWeight(float weight) {
        this.weight=weight;
    }
    public float getHeight() {
        return height;
    }
    public void setHeight(float height) {
        this.height=height;
    }
    public void acceptRecord() {
        Scanner sc = new Scanner(System.in);
        setWeight(sc.nextFloat());
        setHeight(sc.nextFloat());
        sc.close();
    }
    public void calculateBMI() {
        bmiCalculate = weight / (height * height);
    }
}
```

```

System.out.println(bmiCalculate);
}
public void classifyBMI() {
if (bmiCalculate < 18.5) {
System.out.println("Underweight");
} else if (18.5 <= bmiCalculate && bmiCalculate < 24.9) {
System.out.println("Normal weight");
} else if (25 <= bmiCalculate && bmiCalculate < 29.9) {
System.out.println("Overweight");
} else if (bmiCalculate >= 30) {
System.out.println("Obese");
} else {
System.out.println("Invalid input!");
}
}
public void printRecord() {
System.out.println("BMI is: " + bmiCalculate);
}
}
public class Q3 {
public static void main(String[] args) {
BMITracker bmi = new BMITracker();
bmi.acceptRecord();
bmi.calculateBMI();
bmi.classifyBMI();
bmi.printRecord();
}
}

```

Output:

```

72
1.52
31.163435
Obese
BMI is: 31.163435

```

#### 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 Assignment_4;
import java.util.Scanner;
class DiscountCalculator{
private float originalPrice;
private float discountRate;
private float discountAmount;
private float finalPrice;
public float getOriginalPrice() {
return originalPrice;
}
public void setOriginalPrice(float originalPrice) {
this.originalPrice=originalPrice;
}
public float getDiscountRate() {
return discountRate;
}
public void setDiscountRate(float discountRate) {
this.discountRate=discountRate;
}
public void acceptRecord(/* DiscountCalculator this*/) {
Scanner sc=new Scanner(System.in);
setOriginalPrice(sc.nextFloat());
setDiscountRate(sc.nextFloat());
sc.close();
}
public void calculateDiscount () {
discountAmount = (float) (originalPrice * (discountRate / 100));
finalPrice = (float)(originalPrice - discountAmount);
}
public void printRecord() {
System.out.println("Discount Amount is: "+discountAmount);
System.out.println("Final price is: "+finalPrice);
}
}
public class Q4 {
public static void main(String[] args) {
```

```
DiscountCalculator dc=new DiscountCalculator();
dc.acceptRecord();
dc.calculateDiscount();
dc.printRecord();
}
}
```

Output:

```
13200
3.3
Discount Amount is: 435.6
Final price is: 12764.4
```



## 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 Assignment_4;
import java.util.Scanner;
public class Q5 {
private double carRate;
private double truckRate;
private double motorcycleRate;
private int carCount;
private int truckCount;
private int motorcycleCount;
// Constructor
public Q5() {
this.carRate = 0.0;
```



```

this.truckRate = 0.0;
this.motorcycleRate = 0.0;
this.carCount = 0;
this.truckCount = 0;
this.motorcycleCount = 0;
}
// Set toll rates for different vehicle types
public void setTollRates(double carRate, double truckRate, double motorcycleRate) {
this.carRate = carRate;
this.truckRate = truckRate;
this.motorcycleRate = motorcycleRate;
}
// Accept the number of vehicles passing through
public void acceptRecord(int carCount, int truckCount, int motorcycleCount) {
this.carCount = carCount;
this.truckCount = truckCount;
this.motorcycleCount = motorcycleCount;
}
// Calculate the total revenue
public double calculateRevenue() {
double totalRevenue = (carCount * carRate) + (truckCount * truckRate) + (motorcycleCount
* motorcycleRate);
return totalRevenue;
}
// Print the record
public void printRecord() {
double totalRevenue = calculateRevenue();
int totalVehicles = carCount + truckCount + motorcycleCount;
System.out.printf("Total Vehicles: %d\n", totalVehicles);
System.out.printf("Total Revenue: ₹%.2f\n", totalRevenue);
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
Q5 manager = new TollBoothRevenueManager();
// Set toll rates
System.out.println("Enter toll rates for Car, Truck, and Motorcycle:");
System.out.print("Car rate: ₹");
double carRate = scanner.nextDouble();
System.out.print("Truck rate: ₹");
double truckRate = scanner.nextDouble();
System.out.print("Motorcycle rate: ₹");
double motorcycleRate = scanner.nextDouble();
manager.setTollRates(carRate, truckRate, motorcycleRate);
// Accept number of vehicles
System.out.println("Enter the number of vehicles:");
System.out.print("Number of Cars: ");
int carCount = scanner.nextInt();

```

```
System.out.print("Number of Trucks: ");  
int truckCount = scanner.nextInt();  
System.out.print("Number of Motorcycles: ");  
int motorcycleCount = scanner.nextInt();  
manager.acceptRecord(carCount, truckCount, motorcycleCount);  
// Print the record  
manager.printRecord();  
// Close scanner  
scanner.close();  
}  
}
```

```
terminated: C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe  
Enter the original price: 5000  
Enter the discount rate (%): 4.3  
Discount Amount is: ₹215.00  
Final Price is: ₹4785.00
```