**Note:**

* The assignment is designed to practice class, fields, and methods only.
* Create a separate project for each question.
* Do not use getter/setter methods or constructors for these assignments.
* Define two classes: one class to implement the logic and another class to test it.

**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 (₹).

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

import java.util.Scanner;

import java.lang.Math;

class LoanAmortizationCalculator {

private float principal;

private float rate;

private int term;

double monthlyPayment;

double totalAmountPaid;

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

public void acceptRecord() {

System.out.print("Enter the principal amount: ");

principal = sc.nextFloat();

System.out.print("Enter the rate of interest: ");

rate = sc.nextFloat();

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

term = sc.nextInt();

sc.close();

}

public void calculateMonthlyPayment() {

monthlyPayment = principal \*

((rate/1200) \*

Math.pow((1 + (rate/1200)),((float)term)\*12))

/ (Math.pow((1 + (rate/1200)),((float)term)\*12) - 1);

totalAmountPaid = monthlyPayment \* term \* 12;

}

public void printRecord() {

System.out.println("Monthly payment = " + monthlyPayment);

System.out.println("Total Amount paid over the life time of loan = " + totalAmountPaid);

}

public static void main(String[] args) {

// TODO Auto-generated method stub

LoanAmortizationCalculator c = new LoanAmortizationCalculator();

c.acceptRecord();

c.calculateMonthlyPayment();

c.printRecord();

//c.sc.close();

}

}

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

import java.util.Scanner;

class CompoundInterestCalculator {

private float principal;

private float rate;

private float compoundingTime;

private float duration;

private double futureValue;

private double totalInterest;

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

public void acceptRecord() {

System.*out*.print("Enter principal amount: ");

principal = *sc*.nextFloat();

System.*out*.print("Enter rate: ");

rate = *sc*.nextFloat();

System.*out*.print("Enter compounding time: ");

compoundingTime = *sc*.nextFloat();

System.*out*.print("Enter duration: ");

duration = *sc*.nextFloat();

}

public void calculateFutureValue() {

futureValue = principal\*Math.*pow*((1+(rate/100)/compoundingTime),

(compoundingTime\*duration));

totalInterest = futureValue - principal;

}

public void printRecord() {

System.*out*.println(futureValue);

System.*out*.println(totalInterest);

}

}

public class CompoundInterestCalculator1{

public static void main(String[] args) {

CompoundInterestCalculator c = new CompoundInterestCalculator();

c.acceptRecord();

c.calculateFutureValue();

c.printRecord();

CompoundInterestCalculator.*sc*.close();

}

}

**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
   * Obese: BMI ≥ 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.

import java.util.Scanner;

class BMITracker{

private float weight;

private float height;

private float bmi;

private String classification;

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

public void acceptRecord(){

System.*out*.print("Enter your weight: ");

weight = sc.nextFloat();

System.*out*.print("Enter your height: ");

height = sc.nextFloat();

}

public void calculateBMI() {

bmi=weight/(height\*height);

}

public void classifyBMI() {

if(bmi<18.5) {

classification = "Underweight";

}

else if(bmi>=18.5 && bmi<24.9){

classification = "Normal weight";

}

else if(bmi>=25 && bmi<29.9f){

classification = "Overweight";

}

else{

classification = "Obese";

}

}

public void printRecord() {

System.*out*.println("Your BMI is: " + bmi);

System.*out*.println("Your are: " + classification);

}

}

public class BMI {

public static void main(String[] args) {

// TODO Auto-generated method stub

BMITracker bt = new BMITracker();

bt.acceptRecord();

bt.calculateBMI();

bt.classifyBMI();

bt.printRecord();

}

}

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

**import** java.util.Scanner;

**class** DiscountCalculator{

**private** **float** originalPrice;

**private** **float** discountRate;

**private** **float** discountAmount;

**private** **float** finalPrice;

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

**public** **void** acceptRecord(){

System.***out***.print("Enter the original price: ");

**this**.originalPrice = sc.nextFloat();

System.***out***.print("Enter the discount percentage: ");

**this**.discountRate = sc.nextFloat();

}

**public** **void** calculateDiscount() {

**this**.discountAmount = originalPrice \* (discountRate / 100);

**this**.finalPrice = originalPrice - discountAmount;

}

**public** **void** printRecord() {

System.***out***.println("Amount after discount = "+ **this**.discountAmount);

System.***out***.println("FInal price = "+ **this**.finalPrice);

}

}

**public** **class** Discount {

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

DiscountCalculator dc = **new** DiscountCalculator();

dc.acceptRecord();

dc.calculateDiscount();

dc.printRecord();

dc.sc.close();

}

}

**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.

import java.util.Scanner;

import java.text.NumberFormat;

import java.util.Locale;

class Toll

{

public int ch;

public int car\_rate;

public int truck\_rate;

public int motorcycle\_rate;

public int a=1;

public int c\_count=0;

public int t\_count=0;

public int m\_count=0;

public int c\_sum=0;

public int t\_sum=0;

public int m\_sum=0;

public int total=0;

public String vname;

Scanner sc =new Scanner (System.in);

public Toll()

{

System.out.println("Default rates if you do not enter rates");

this.car\_rate=50;

this.truck\_rate=100;

this.motorcycle\_rate=30;

System.out.println("Car :"+this.car\_rate);

System.out.println("Truck :"+this.truck\_rate);

System.out.println("Motorcycle :"+this.motorcycle\_rate);

}

public void setTollRates()

{

System.out.println("Do you Want to Enter The New Toll Prices ? \n1.Yes\n2.No");

a =sc.nextInt();

if(a==1)

{

System.out.println("Enter Toll Rate For Car:");

this.car\_rate=sc.nextInt();

System.out.println("Enter Toll Rate For Truck:");

this.truck\_rate=sc.nextInt();

System.out.println("Enter Toll Rate For Motorcycle:");

this.motorcycle\_rate=sc.nextInt();

}

}

public void accept\_record()

{

do {

System.out.println("Select Your Vehicle Type:\n1.Car\n2.Truck\n3.Motorcycle");

ch=sc.nextInt();

switch (ch) {

case 1:

{

c\_sum=c\_sum+car\_rate;

c\_count++;

}

break;

case 2:

{

t\_sum=t\_sum+truck\_rate;

t\_count++;

}

break;

case 3:

{

m\_sum=m\_sum+motorcycle\_rate;

m\_count++;

}

break;

default:

System.out.println("Invalid input");

break;

}

System.out.println("Do you Want to add vehicle ? \n1.Yes\n2.No");

a =sc.nextInt();

} while (a!=2);

}

public void calculateRevenue()

{

total=m\_sum+t\_sum+c\_sum;

}

public void printRecord()

{

//NumberFormat india = NumberFormat.getCurrencyInstance(new Locale("en","in"));

System.out.println("Car : "+c\_count+" x "+car\_rate+" = "+c\_sum);

System.out.println("Truck : "+t\_count+" x "+truck\_rate+" = "+t\_sum);

System.out.println("MotorCycle : "+m\_count+" x "+motorcycle\_rate+" = "+m\_sum);

System.out.println("------------------------------");

System.out.println("Total Revenu: "+total);

System.out.println("Total count: "+ (int)(c\_count+t\_count+m\_count));

}

}

public class TollBooth {

public static void main(String[] args) {

Toll w =new Toll();

w.setTollRates();

w.accept\_record();

w.calculateRevenue();

w.printRecord();

}

}