Note:

* The assignment is designed to practice constructor, getter/setter and toString method.
* Create a separate project for each question and create separate file for each class.
* Try to test the functionality by using menu-driven program.

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 the class LoanAmortizationCalculator with fields, an appropriate constructor, getter and setter methods, a toString method and business logic methods. Define the class LoanAmortizationCalculatorUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method and test the functionality of the utility class.

*AcceptRecord.java*

package org.example.loan;

public class AcceptRecord {

float loanAmm;

float anInt;

float loanTerm;

float monthlyPay;

float monthlyInt;

float noOfMonths;

float totalAm;

public float getLoanAmm() {

return loanAmm;

}

public void setLoanAmm(float loanAmm) {

this.loanAmm = loanAmm;

}

public float getAnInt() {

return anInt;

}

public void setAnInt(float anInt) {

this.anInt = anInt;

}

public float getLoanTerm() {

return loanTerm;

}

public void setLoanTerm(float loanTerm) {

this.loanTerm = loanTerm;

}

public float getMonthlyInt() {

return monthlyInt;

}

public void setMonthlyInt(float monthlyInt) {

this.monthlyInt = monthlyInt;

}

public float getNoOfMonths() {

return noOfMonths;

}

public void setNoOfMonths(float noOfMonths) {

this.noOfMonths = noOfMonths;

}

public void calcMonthlyPay() {

monthlyInt = (float) ((Math.*pow*((1.0+(anInt/100)), 1.0/12.0)) - 1.0);

noOfMonths = loanTerm\*12;

monthlyPay = (float) ((loanAmm \* monthlyInt)/(1.0 - (Math.*pow*((1.0+monthlyInt), -noOfMonths))));

totalAm = monthlyPay \* noOfMonths;

}

*@Override*

public String toString() {

return "Monthly payment of load: " + monthlyPay + ", Total ammount to be paid: " + totalAm;

}

}

*Program.java*

package org.example.loan;

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

AcceptRecord rec = null;

rec = new AcceptRecord( );

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

System.***out***.print("Enter loan ammount: ");

rec.setLoanAmm(sc.nextFloat());

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

rec.setAnInt(sc.nextFloat());

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

rec.setLoanTerm(sc.nextFloat());

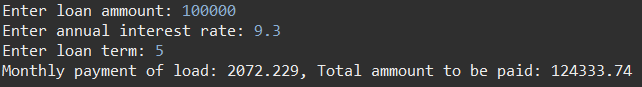
rec.calcMonthlyPay();

System.***out***.println(rec.toString());

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 the class CompoundInterestCalculator with fields, an appropriate constructor, getter and setter methods, a toString method and business logic methods. Define the class CompoundInterestCalculatorUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

package org.example.comp.interest;

import java.util.Scanner;

class Calculator{

float initInvAmm, annInt, noOfCompounded, dur;

double futureValue, totalInterest;

public float getInitInvAmm() {

return initInvAmm;

}

public void setInitInvAmm(float initInvAmm) {

this.initInvAmm = initInvAmm;

}

public float getAnnInt() {

return annInt;

}

public void setAnnInt(float annInt) {

this.annInt = annInt;

}

public float getNoOfCompounded() {

return noOfCompounded;

}

public void setNoOfCompounded(float noOfCompounded) {

this.noOfCompounded = noOfCompounded;

}

public float getDur() {

return dur;

}

public void setDur(float dur) {

this.dur = dur;

}

public void calculateTotalEarn() {

futureValue = (initInvAmm \* Math.*pow*(((1 + annInt)/noOfCompounded), (noOfCompounded\*dur)));

totalInterest = futureValue - initInvAmm;

}

*@Override*

public String toString() {

return "Future value is : " + futureValue + " Total interest earned is : " + totalInterest;

}

}

public class Program {

public static void main(String[] args) {

Calculator cl = new Calculator();

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

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

cl.setInitInvAmm(sc.nextFloat());

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

cl.setAnnInt(sc.nextFloat());

System.***out***.println("Enter number of times the interest is compounded per year: ");

cl.setNoOfCompounded(sc.nextFloat());

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

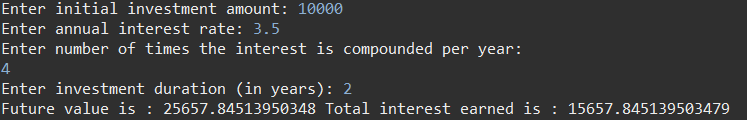
cl.setDur(sc.nextFloat());

cl.calculateTotalEarn();

System.***out***.println(cl.toString());

}

}



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 the class BMITracker with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic methods. Define the class BMITrackerUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

package org.example.bmi;

import java.util.Scanner;

class BMITracker {

float height, weight, bmi;

public float getHeight() {

return height;

}

public void setHeight(float height) {

this.height = height;

}

public float getWeight() {

return weight;

}

public void setWeight(float weight) {

this.weight = weight;

}

public String calculateBMI() {

bmi = weight / (height \* height);

return this.toString();

}

public String evaluateBMI() {

String status;

if(bmi < 18.5) {

status = "Underweight";

}

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

status = "Normal weight";

}

else if(bmi >= 25.0 && bmi < 29.9) {

status = "Overweight";

}

else {

status = "Obese";

}

return status;

}

*@Override*

public String toString() {

return "BMI status is : " + this.evaluateBMI() + ".";

}

}

public class Program {

public static void main(String[] args) {

BMITracker bt = new BMITracker();

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

System.***out***.println("Enter weight (in kilograms): ");

bt.setWeight(sc.nextLong());

System.***out***.println("Enter height (in meters) : ");

bt.setHeight(sc.nextFloat());

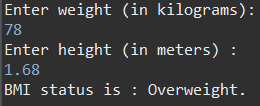
System.***out***.println(bt.calculateBMI());

// bt.toString();

sc.close();

}

}



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 the class DiscountCalculator with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic methods. Define the class DiscountCalculatorUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

package org.example.discount;

import java.util.Scanner;

class DiscountCalculator{

float ogPrice, dcPerc;

float dcAmmount, finalPrice;

public float getOgPrice() {

return ogPrice;

}

public void setOgPrice(float ogPrice) {

this.ogPrice = ogPrice;

}

public float getDcPerc() {

return dcPerc;

}

public void setDcPerc(float dcPerc) {

this.dcPerc = dcPerc;

}

public void calculateDiscount() {

dcAmmount = ogPrice \* (dcPerc/100);

finalPrice = ogPrice - dcAmmount;

}

*@Override*

public String toString() {

return "Discounted ammount is : " + dcAmmount + " Final price is : " + finalPrice;

}

}

public class Program {

public static void main(String[] args) {

DiscountCalculator cl = new DiscountCalculator();

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

System.***out***.println("Enter Original Ammount: ");

cl.setOgPrice(sc.nextFloat());

System.***out***.println("Enter Discount Percentage: ");

cl.setDcPerc(sc.nextFloat());

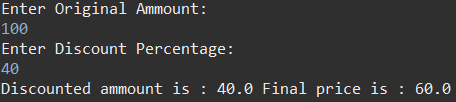
cl.calculateDiscount();

System.***out***.println(cl.toString());

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 the class TollBoothRevenueManager with fields, an appropriate constructor, getter and setter methods, a toString method, and business logic methods. Define the class TollBoothRevenueManagerUtil with methods acceptRecord, printRecord, and menuList. Define the class Program with a main method to test the functionality of the utility class.

package org.example.revevnue;

import java.util.Scanner;

class TollBoothRevenueManager{

float carRate, truckRate, motoRate, totalRev;

int noOfCars, noOfTrucks, noOfMoto, totalVeh;

public float getCarRate() {

return carRate;

}

public void setCarRate(float carRate) {

this.carRate = carRate;

}

public float getTruckRate() {

return truckRate;

}

public void setTruckRate(float truckRate) {

this.truckRate = truckRate;

}

public float getMotoRate() {

return motoRate;

}

public void setMotoRate(float motoRate) {

this.motoRate = motoRate;

}

public float getTotalRev() {

return totalRev;

}

public void setTotalRev(float totalRev) {

this.totalRev = totalRev;

}

public int getNoOfCars() {

return noOfCars;

}

public void setNoOfCars(int noOfCars) {

this.noOfCars = noOfCars;

}

public int getNoOfTrucks() {

return noOfTrucks;

}

public void setNoOfTrucks(int noOfTrucks) {

this.noOfTrucks = noOfTrucks;

}

public int getNoOfMoto() {

return noOfMoto;

}

public void setNoOfMoto(int noOfMoto) {

this.noOfMoto = noOfMoto;

}

public int getTotalVeh() {

return totalVeh;

}

public void setTotalVeh(int totalVeh) {

this.totalVeh = totalVeh;

}

public void calculateRevenue() {

totalVeh = noOfCars + noOfTrucks + noOfMoto;

totalRev = (noOfCars\*carRate) + (noOfTrucks\*noOfCars) + (noOfMoto\*motoRate);

}

*@Override*

public String toString() {

return "Total vehicle are : " + totalVeh + " Total revanue is : " + totalRev;

}

}

public class Program {

public static void main(String[] args) {

TollBoothRevenueManager rm = new TollBoothRevenueManager();

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

System.***out***.println("Enter toll rate for cars: ");

rm.setCarRate(sc.nextFloat());

System.***out***.println("Enter toll rate for truck: ");

rm.setTruckRate(sc.nextFloat());

System.***out***.println("Enter toll rate for motorcycle: ");

rm.setMotoRate(sc.nextFloat());

System.***out***.println("Enter no. of cars: ");

rm.setNoOfCars(sc.nextInt());

System.***out***.println("Enter no. of truck: ");

rm.setNoOfTrucks(sc.nextInt());

System.***out***.println("Enter no. of motorcycle: ");

rm.setNoOfMoto(sc.nextInt());

rm.calculateRevenue();

System.***out***.println(rm.toString());

sc.close();

}

}

