**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.

**LoanAmortizationCalculator.java**

**package** ass4\_1.example;

**public** **class** LoanAmortizationCalculator {

**float** principal;

**float** rate;

**int** years;

**double** monthlyPayment;

**int** n;

**double** totAmount;

**public** LoanAmortizationCalculator(**float** principal, **float** rate, **int** years) {

**this**.principal = principal;

**this**.rate = rate;

**this**.years = years;

}

**public** **float** getPrincipal() {

**return** principal;

}

**public** **void** setPrincipal(**float** principal) {

**this**.principal = principal;

}

**public** **float** getRate() {

**return** rate;

}

**public** **void** setRate(**float** rate) {

**this**.rate = rate;

}

**public** **int** getYears() {

**return** years;

}

**public** **void** setYears(**int** years) {

**this**.years = years;

}

**public** **double** calculateMonthlyPayment() {

**double** monthlyInterestRate= rate/12/100;

n=years\*12;

monthlyPayment = principal\*(monthlyInterestRate\*(Math.*pow*((1+monthlyInterestRate),n)))/(Math.*pow*((1+monthlyInterestRate),n)-1);

**return** monthlyPayment;

}

**public** **double** calTotalAmount() {

// **TODO** Auto-generated method stub

totAmount= monthlyPayment\*n;

**return** totAmount;

}

// toString method to display loan details

@Override

**public** String toString() {

**return** "Monthly Payment: Rs. " + calculateMonthlyPayment()+"\n"+

"Total Amount Paid "+calTotalAmount();

}

}

**LoanAmortizationCalculatorUtil.java**

**package** ass4\_1.example;

**import** java.util.Scanner;

**public** **class** LoanAmortizationCalculatorUtil {

**private** LoanAmortizationCalculator loan;

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

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

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

**float** principal = sc.nextFloat();

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

**float** rate=sc.nextFloat();

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

**int** years=sc.nextInt();

loan = **new** LoanAmortizationCalculator(principal, rate, years);

}

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

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

}

// Menu-driven program

**public** **void** menuList() {

**while** (**true**) {

System.***out***.println("\nMenu:");

System.***out***.println("1. acceptRecord");

System.***out***.println("2. Print Record");

System.***out***.println("3. Exit");

System.***out***.print("Choose an option: ");

**int** choice = sc.nextInt();

**switch** (choice) {

**case** 1:

acceptRecord();

**break**;

**case** 2:

printRecord();

**break**;

**case** 3:

System.*exit*(0);

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

}

}

**Program.java**

package ass4\_1.example;

public class Program {

public static void main(String[] args) {

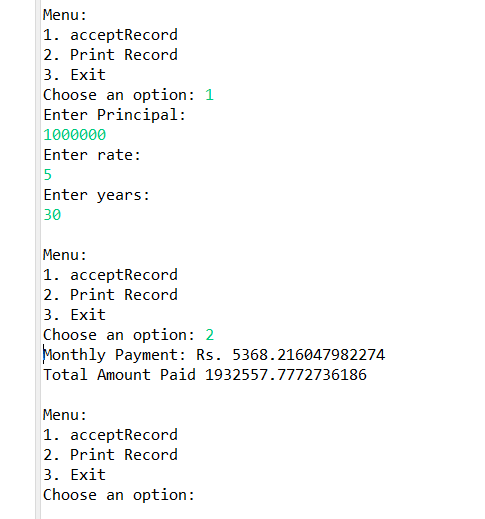
// TODO Auto-generated method stub

LoanAmortizationCalculatorUtil util = new LoanAmortizationCalculatorUtil();

util.menuList();

}

}

****

**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.

**CompoundInterestCalculator.java**

package ass4\_2;

public class CompoundInterestCalculator {

private double principal;

private double annualInterest;

private int noOfCompound;

private int years;

private double futureValue;

private double totInterest;

public CompoundInterestCalculator(double principal, double annualInterest, int noOfCompound, int years) {

this.principal = principal;

this.annualInterest = annualInterest;

this.noOfCompound = noOfCompound;

this.years = years;

}

public double getPrincipal() {

return principal;

}

public void setPrincipal(double principal) {

this.principal = principal;

}

public double getAnnualInterest() {

return annualInterest;

}

public void setAnnualInterest(double annualInterest) {

this.annualInterest = annualInterest;

}

public int getNoOfCompound() {

return noOfCompound;

}

public void setNoOfCompound(int noOfCompound) {

this.noOfCompound = noOfCompound;

}

public int getYears() {

return years;

}

public void setYears(int years) {

this.years = years;

}

public double calculateFutureValue() {

futureValue= principal \*Math.*pow* ((1 + annualInterest / noOfCompound),noOfCompound \* years);

return futureValue;

}

public double totalInterest()

{

totInterest=futureValue-principal;

return totInterest;

}

@Override

public String toString() {

return "Future Value= " + calculateFutureValue() + ", Total Interest= "

+ totalInterest();

}

}

**CompoundInterestCalculatorUtil.java**

**package** ass4\_2;

**import** java.util.Scanner;

**public** **class** CompoundInterestCalculatorUtil {

**private** CompoundInterestCalculator ci;

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

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

System.***out***.println("Initial Investment Ammount:");

**double** principal = sc.nextDouble();

System.***out***.println("Annual Interest Rate:");

**double** annualInterest = sc.nextDouble();

System.***out***.println("Number of time interest compounded:");

**int** noOfCompound = sc.nextInt();

System.***out***.println("Enter year:");

**int** years = sc.nextInt();

ci=**new** CompoundInterestCalculator(principal, annualInterest,noOfCompound , years);

}

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

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

}

**public** **void** menuList() {

**while** (**true**) {

System.***out***.println("\nMenu:");

System.***out***.println("1. acceptRecord");

System.***out***.println("2. Print Record");

System.***out***.println("3. Exit");

System.***out***.print("Choose an option: ");

**int** choice = sc.nextInt();

**switch** (choice) {

**case** 1:

acceptRecord();

**break**;

**case** 2:

printRecord();

**break**;

**case** 3:

System.*exit*(0);

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

}

}

**Program.java**

package ass4\_2;

public class Program {

public static void main(String[] args) {

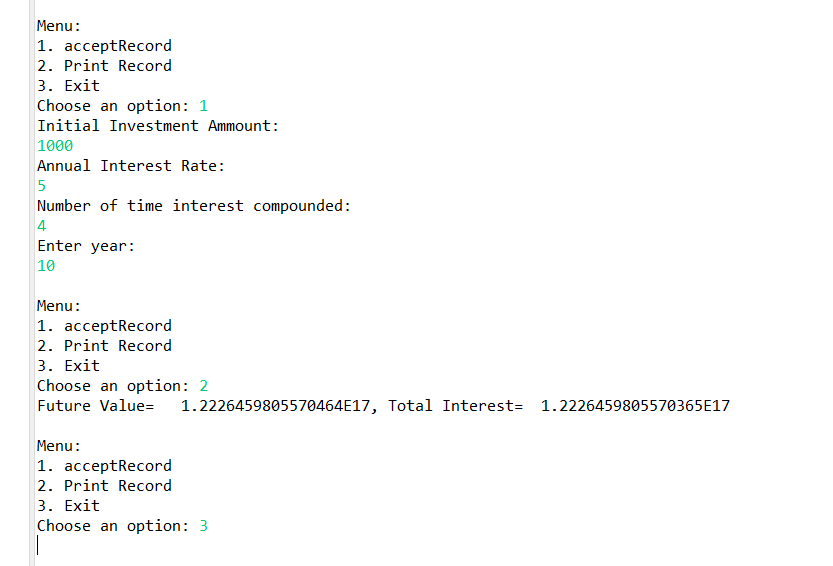
// TODO Auto-generated method stub

CompoundInterestCalculatorUtil util=new CompoundInterestCalculatorUtil();

util.menuList();

}

}

****

**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.

**BMITracker.java**

**package** ass4\_3.example;

**public** **class** BMITracker {

**private** **float** weight;

**private** **float** height;

**static** **float** *bmi*;

**public** **float** calculateBMI() {

*bmi*=weight/(weight\*height);

**return** *bmi*;

}

**public** BMITracker(**float** weight, **float** height) {

**this**.weight = weight;

**this**.height = height;

}

**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;

}

@Override

**public** String toString() {

**return** "BMI =" + calculateBMI() ;

}

}

**BMITrackerUtil.java**

package ass4\_3.example;

import java.util.Scanner;

public class BMITrackerUtil {

private BMITracker b;

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

public void acceptRecord() {

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

float weight = sc.nextFloat();

System.*out*.println("Enter height");

float height = sc.nextFloat();

b=new BMITracker(weight,height);

}

public void printRecord () {

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

if(BMITracker.*bmi*<18.5)

System.*out*.println("Underweight");

else if(BMITracker.*bmi*>=18.5 && BMITracker.*bmi*<=24.9)

System.*out*.println("Normal weight");

else if (BMITracker.*bmi*>=25 && BMITracker.*bmi*<=29.9)

System.*out*.println("Overweight");

else

System.*out*.println("Obese");

}

public void menuList() {

while (true) {

System.*out*.println("\nMenu:");

System.*out*.println("1. acceptRecord");

System.*out*.println("2. Print Record");

System.*out*.println("3. Exit");

System.*out*.print("Choose an option: ");

int choice = sc.nextInt();

switch (choice) {

case 1:

acceptRecord();

break;

case 2:

printRecord();

break;

case 3:

System.*exit*(0);

break;

default:

System.*out*.println("Invalid choice. Please try again.");

}

}

}

}

**Program.java**

package ass4\_3.example;

public class Program {

public static void main(String[] args) {

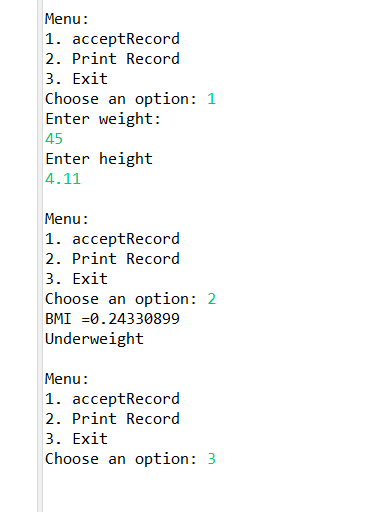
// TODO Auto-generated method stub

BMITrackerUtil util=new BMITrackerUtil();

util.menuList();

}

}

****

**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.

**DiscountCalculator.java**

**package ass4\_4;**

**public class DiscountCalculator {**

**private float price;**

**private int discount;**

**private float disAmount;**

**private float finalPrice;**

**public DiscountCalculator(float price, int discount) {**

**this.price = price;**

**this.discount = discount;**

**}**

**public float getPrice() {**

**return price;**

**}**

**public void setPrice(float price) {**

**this.price = price;**

**}**

**public int getDiscount() {**

**return discount;**

**}**

**public void setDiscount(int discount) {**

**this.discount = discount;**

**}**

**public double calculateDiscount() {**

**disAmount= price\* discount/100;**

**return disAmount;**

**}**

**public double finalPrice()**

**{**

**finalPrice=price-disAmount;**

**return finalPrice;**

**}**

**@Override**

**public String toString() {**

**return "Discount=" + calculateDiscount() + ", FinalPrice=" + finalPrice();**

**}**

**}**

**DiscountCalculatorUtil.java**

**package ass4\_4;**

**import java.util.Scanner;**

**public class DiscountCalculatorUtil {**

**private DiscountCalculator dis;**

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

**public void acceptRecord() {**

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

**float price = sc.nextFloat();**

**System.*out*.println("Enter discount: ");**

**int discount = sc.nextInt();**

**dis=new DiscountCalculator(price,discount);**

**}**

**public void printRecord() {**

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

**}**

**public void menuList() {**

**while (true) {**

**System.*out*.println("\nMenu:");**

**System.*out*.println("1. acceptRecord");**

**System.*out*.println("2. Print Record");**

**System.*out*.println("3. Exit");**

**System.*out*.print("Choose an option: ");**

**int choice = sc.nextInt();**

**switch (choice) {**

**case 1:**

**acceptRecord();**

**break;**

**case 2:**

**printRecord();**

**break;**

**case 3:**

**System.*exit*(0);**

**break;**

**default:**

**System.*out*.println("Invalid choice. Please try again.");**

**}**

**}**

**}**

**}**

**Program.java**

**package ass4\_4;**

**public class Program {**

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

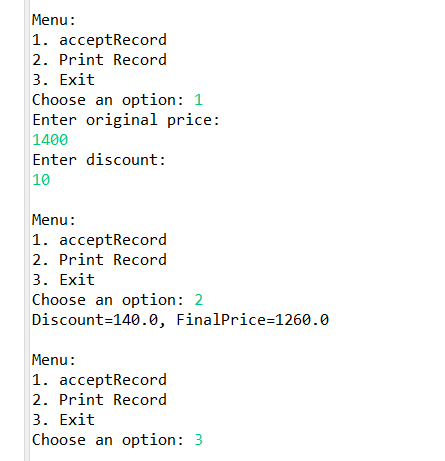
**{**

**DiscountCalculatorUtil util=new DiscountCalculatorUtil();**

**util.menuList();**

**}**

**}**

****

**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.