**ASSIGNMENT 4**

package com.example;

import java.util.Scanner;

/\*

• The assignment is designed to practice constructor, getter/setter and toString method.

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:

o 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.

\*/

class LoanAmortizationCalculator{

private int principle\_amount;

private float interest\_rate;

private float loan\_term;

public LoanAmortizationCalculator() {

}

public LoanAmortizationCalculator(int principle\_amount, float interest\_rate, float loan\_term) {

this.principle\_amount = principle\_amount;

this.interest\_rate = interest\_rate;

this.loan\_term = loan\_term;

}

public int getPrinciple\_amount() {

return principle\_amount;

}

public void setPrinciple\_amount(int principle\_amount) {

this.principle\_amount = principle\_amount;

}

public float getInterest\_rate() {

return interest\_rate;

}

public void setInterest\_rate(float interest\_rate) {

this.interest\_rate = interest\_rate;

}

public float getLoan\_term() {

return loan\_term;

}

public void setLoan\_term(float loan\_term) {

this.loan\_term = loan\_term;

}

public void acceptRecord() {

// TODO Auto-generated method stub

}

public void printRrcord() {

// TODO Auto-generated method stub

}

public double calculateMonthlyPayment() {

double monthlyInterestRate = (interest\_rate / 12) / 100;

double numberOfMonths = loan\_term \* 12;

double monthlyPayment = principle\_amount \* (monthlyInterestRate \* Math.*pow*(1 + monthlyInterestRate, numberOfMonths))

/ (Math.*pow*(1 + monthlyInterestRate, numberOfMonths) - 1);

System.*out*.println( "monthly payment :" + monthlyPayment);

return monthlyPayment;

}

public double calculateAnnualPayment() {

double calculateMonthly = calculateMonthlyPayment();

double numberOfMonths = loan\_term \* 12;

double annualPayment = numberOfMonths \* calculateMonthly ;

System.*out*.println(annualPayment);

return annualPayment;

}

}

class LoanAmortizationCalculatorUtil{

LoanAmortizationCalculator lc=new LoanAmortizationCalculator();

void acceptRecord() {

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

System.*out*.println("enter the principle amount:");

int principle=sc.nextInt();

lc.setPrinciple\_amount(principle);

System.*out*.println("enter the interest rate:");

float interest\_rate=sc.nextFloat();

lc.setInterest\_rate(interest\_rate);

System.*out*.println("enter tha loan term:");

float loan\_term=sc.nextFloat();

lc.setLoan\_term(loan\_term);

System.*out*.println();

//appMenuList();

}

void printRecord() {

System.*out*.println("principle amount is:" + lc.getPrinciple\_amount());

System.*out*.println("interest rate is:" + lc.getInterest\_rate());

System.*out*.println("loan term is:"+ lc.getLoan\_term());

}

void appMenuList() {

System.*out*.println("1. Display Record");

System.*out*.println("2. Accept Records");

System.*out*.println("3. Monthly Payment");

System.*out*.println("4. Yearly Payment");

System.*out*.println("5. exit");

System.*out*.println();

System.*out*.println("Please enter number to proceed");

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

int choise = scanner.nextInt();

switch(choise) {

case 1 :

printRecord();

appMenuList();

break;

case 2 :

acceptRecord();

appMenuList();

break;

case 3 :

lc.calculateMonthlyPayment();

appMenuList();

break;

case 4 :

lc.calculateAnnualPayment();

appMenuList();

break;

case 5:

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

break;

default :

System.*out*.println("please enter valid choise");

appMenuList();

break;

}

}

}

public class Test {

public static void main(String[] args) {

System.*out*.println("Welcomen to CDAC Loan calculator");

System.*out*.println();

LoanAmortizationCalculatorUtil amortizationCalculatorUtil = new LoanAmortizationCalculatorUtil();

amortizationCalculatorUtil.appMenuList();

}

}

**package** com.cdac;

**import** java.util.Scanner;

/\*

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:

o Future Value Calculation:

♣ futureValue = principal \* (1 + annualInterestRate / numberOfCompounds)^(numberOfCompounds \* years)

o 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.

\*/

**class** CompundInterestCalculator {

**private** **int** initial\_investment\_amount;

**private** **float** interest\_rate;

**private** **float** no\_of\_times\_componded;

**private** **float** investment\_duration;

**double** futureValue;

**double** totalInterest;

**public** **int** getInitial\_investment\_amount() {

**return** initial\_investment\_amount;

}

**public** **void** setInitial\_investment\_amount(**int** initial\_investment\_amount) {

**this**.initial\_investment\_amount = initial\_investment\_amount;

}

**public** **float** getInterest\_rate() {

**return** interest\_rate;

}

**public** **void** setInterest\_rate(**float** interest\_rate) {

**this**.interest\_rate = interest\_rate;

}

**public** **float** getNo\_of\_times\_componded() {

**return** no\_of\_times\_componded;

}

**public** **void** setNo\_of\_times\_componded(**float** no\_of\_times\_componded) {

**this**.no\_of\_times\_componded = no\_of\_times\_componded;

}

**public** **float** getInvestment\_duration() {

**return** investment\_duration;

}

**public** **void** setInvestment\_duration(**float** investment\_duration) {

**this**.investment\_duration = investment\_duration;

}

**public** CompundInterestCalculator() {

}

**void** futureValue() {

futureValue = initial\_investment\_amount

\* Math.*pow*((1 + interest\_rate / no\_of\_times\_componded), (no\_of\_times\_componded \* investment\_duration));

System.***out***.println(futureValue);

}

**void** totalInterest() {

totalInterest = futureValue - initial\_investment\_amount;

System.***out***.println(totalInterest);

}

}

**class** CompoundInterestCalculatorUti {

CompundInterestCalculator compoundinterestcal = **new** CompundInterestCalculator();

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

**void** acceptRecord() {

System.***out***.println("enter the initial investment:");

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

compoundinterestcal.setInitial\_investment\_amount(principle);

System.***out***.println("enter interest rate");

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

compoundinterestcal.setInterest\_rate(rate);

System.***out***.println("enter the no of times compounded");

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

compoundinterestcal.setNo\_of\_times\_componded(time);

System.***out***.println("enter the duration:");

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

compoundinterestcal.setInvestment\_duration(duration);

}

**void** printRecord()

{

System.***out***.println("principle is:" + compoundinterestcal.getInitial\_investment\_amount());

System.***out***.println("interest rate is:" + compoundinterestcal.getInterest\_rate());

System.***out***.println("no of times compounede is:" + compoundinterestcal.getNo\_of\_times\_componded());

System.***out***.println("the duration is :" + compoundinterestcal.getInvestment\_duration());

}

**void** menuList() {

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

System.***out***.println("1.Accept Record ");

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

System.***out***.println("3.Know Future value ");

System.***out***.println("4.Interest earned");

System.***out***.println("5.exit");

System.***out***.println("enter the choice");

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

**switch** (choice) {

**case** 1:

acceptRecord();

menuList();

**break**;

**case** 2:

printRecord();

menuList();

**break**;

**case** 3:

compoundinterestcal.futureValue();

menuList();

**break**;

**case** 4:

compoundinterestcal.totalInterest();

menuList();

**break**;

**case** 5:

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

**break**;

**default**:

System.***out***.println("enter the valid choice:");

**break**;

}

}

}

**public** **class** Test {

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

System.***out***.println("Welcome to cdac compound interest calculator:");

CompoundInterestCalculatorUti cti = **new** CompoundInterestCalculatorUti();

cti.menuList();

}

}

**package** com.cdac;

**import** java.util.Scanner;

/\*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:

o BMI Calculation: BMI = weight / (height \* height)

3. Classify the BMI into one of the following categories:

o Underweight: BMI < 18.5

o Normal weight: 18.5 ≤ BMI < 24.9

o Overweight: 25 ≤ BMI < 29.9

o 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.

\*/

**public** **class** Test {

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

System.***out***.println("Welcome to BMI calculator:");

BMITrackerUtil btu=**new** BMITrackerUtil();

btu.menuList();

}

}

**class** BMITracker{

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

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

**private** **double** BMI;

**public** BMITracker() {

**super**();

// **TODO** Auto-generated constructor stub

}

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

**super**();

**this**.weight = weight;

**this**.height = height;

BMI = bMI;

}

**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** **double** getBMI() {

**return** BMI;

}

**public** **void** setBMI(**double** bMI) {

BMI = bMI;

}

@Override

**public** String toString() {

**return** "BMITracker [weight=" + weight + ", height=" + height + ", BMI=" + BMI + "]";

}

**public** **void** calculateBMI() {

BMI = weight / (height \* height);

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

}

**void** classify() {

**if** (BMI < 18.5) {

System.***out***.println("you are underweight");

} **else** **if** (BMI >= 18.5 && BMI < 24.9) {

System.***out***.println("you are is normal");

} **else** **if** (BMI >= 25 && BMI < 29.9) {

System.***out***.println("you are overweight");

} **else**

System.***out***.println("you are obese");

}

}

**class** BMITrackerUtil{

BMITracker bt =**new** BMITracker();

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

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

{

System.***out***.println("enter the height");

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

bt.setHeight(height);

System.***out***.println("enter the weight");

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

bt.setWeight(weight);

}

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

{

System.***out***.println("height is " +bt.getHeight());

System.***out***.println("weight is "+bt.getWeight());

}

**void** menuList() {

System.***out***.println("1.Accept Record ");

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

System.***out***.println("3.Know BMI ");

System.***out***.println("4.Classify BMI");

System.***out***.println("5.exit");

System.***out***.println("enter the choice");

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

**switch** (choice) {

**case** 1:

acceptRecord();

menuList();

**break**;

**case** 2:

printRecord();

menuList();

**break**;

**case** 3:

bt.calculateBMI();

menuList();

**break**;

**case** 4:

bt.classify();

menuList();

**break**;

**case** 5:

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

**break**;

**default**:

System.***out***.println("enter the valid choice:");

**break**;

}

}

}

**package** com.cdac;

**import** java.util.Scanner;

/\*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:

o Discount Amount Calculation: discountAmount = originalPrice \* (discountRate / 100)

o 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.

\*/

**public** **class** Test {

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

DiscountCalculatorUtil dtu=**new** DiscountCalculatorUtil();

dtu.menuList();

}

}

**class** DiscountCalculator{

**private** **float** original\_price;

**private** **float** discount;

**private** **double** discount\_amount;

**public** DiscountCalculator() {

}

**public** DiscountCalculator(**float** original\_price, **float** discount) {

**super**();

**this**.original\_price = original\_price;

**this**.discount = discount;

}

**public** **float** getOriginal\_price() {

**return** original\_price;

}

**public** **void** setOriginal\_price(**float** original\_price) {

**this**.original\_price = original\_price;

}

**public** **float** getDiscount() {

**return** discount;

}

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

**this**.discount = discount;

}

@Override

**public** String toString() {

**return** "DiscountCalculator [original\_price=" + original\_price + ", discount=" + discount + "]";

}

**void** discountAmount() {

discount\_amount = original\_price \* (discount / 100);

System.***out***.println(discount\_amount);

}

**void** finalPrice() {

**double** finalPrice = original\_price - discount\_amount;

System.***out***.println(finalPrice);

}

}

**class** DiscountCalculatorUtil{

DiscountCalculator ac=**new** DiscountCalculator();

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

**void** acceptRecord() {

System.***out***.println("enter the original\_price");

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

ac.setOriginal\_price(price);

System.***out***.println("enter the discount");

**float** discount1=sc.nextFloat();

ac.setDiscount(discount1);;

}

**void** printRecord() {

System.***out***.println("original price is"+ac.getOriginal\_price());

System.***out***.println("discount percentege is:"+ac.getDiscount());

}

**void** menuList() {

System.***out***.println("1.Accept Record ");

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

System.***out***.println("3.discount amount ");

System.***out***.println("4.final price");

System.***out***.println("5.exit");

System.***out***.println("enter the choice");

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

**switch** (choice) {

**case** 1:

acceptRecord();

menuList();

**break**;

**case** 2:

printRecord();

menuList();

**break**;

**case** 3:

ac.discountAmount();

menuList();

**break**;

**case** 4:

ac.finalPrice();

menuList();

**break**;

**case** 5:

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

**break**;

**default**:

System.***out***.println("enter the valid choice:");

**break**;

}

}

}

**package** com.cdac;

**import** java.util.Scanner;

/\*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:

o Car: ₹50.00

o Truck: ₹100.00

o 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.

\*/

**public** **class** Test {

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

System.***out***.println("Welcome to cdac toll revenue system:");

TollBoothRevenueManagerUtil tbu=**new** TollBoothRevenueManagerUtil();

tbu.menuList();

}

}

**class** TollBoothRevenueManager{

**private** **int** no\_of\_car;

**private** **int** no\_of\_truck;

**private** **int** no\_of\_motorcycle;

**private** **float** car\_toll;

**private** **float** truck\_toll;

**private** **float** motorcycle\_toll;

**private** **float** total\_revenue;

**public** TollBoothRevenueManager() {

}

**public** TollBoothRevenueManager(**int** no\_of\_car, **int** no\_of\_truck, **int** no\_of\_motorcycle, **float** car\_toll,

**float** truck\_toll, **float** motorcycle\_toll) {

**super**();

**this**.no\_of\_car = no\_of\_car;

**this**.no\_of\_truck = no\_of\_truck;

**this**.no\_of\_motorcycle = no\_of\_motorcycle;

**this**.car\_toll = car\_toll;

**this**.truck\_toll = truck\_toll;

**this**.motorcycle\_toll = motorcycle\_toll;

}

**public** **int** getNo\_of\_car() {

**return** no\_of\_car;

}

**public** **void** setNo\_of\_car(**int** no\_of\_car) {

**this**.no\_of\_car = no\_of\_car;

}

**public** **int** getNo\_of\_truck() {

**return** no\_of\_truck;

}

**public** **void** setNo\_of\_truck(**int** no\_of\_truck) {

**this**.no\_of\_truck = no\_of\_truck;

}

**public** **int** getNo\_of\_motorcycle() {

**return** no\_of\_motorcycle;

}

**public** **void** setNo\_of\_motorcycle(**int** no\_of\_motorcycle) {

**this**.no\_of\_motorcycle = no\_of\_motorcycle;

}

**public** **float** getCar\_toll() {

**return** car\_toll;

}

**public** **void** setCar\_toll(**float** car\_toll) {

**this**.car\_toll = car\_toll;

}

**public** **float** getTruck\_toll() {

**return** truck\_toll;

}

**public** **void** setTruck\_toll(**float** truck\_toll) {

**this**.truck\_toll = truck\_toll;

}

**public** **float** getMotorcycle\_toll() {

**return** motorcycle\_toll;

}

**public** **void** setMotorcycle\_toll(**float** motorcycle\_toll) {

**this**.motorcycle\_toll = motorcycle\_toll;

}

@Override

**public** String toString() {

**return** "TollBoothRevenueManager [no\_of\_car=" + no\_of\_car + ", no\_of\_truck=" + no\_of\_truck

+ ", no\_of\_motorcycle=" + no\_of\_motorcycle + ", car\_toll=" + car\_toll + ", truck\_toll=" + truck\_toll

+ ", motorcycle\_toll=" + motorcycle\_toll + "]";

}

**void** totalVehicles() {

**float** total\_vehicle=**this**.no\_of\_car+**this**.no\_of\_truck+**this**.no\_of\_motorcycle;

System.***out***.println(total\_vehicle);

}

**void** total\_revenue() {

total\_revenue=(no\_of\_car \* car\_toll)+(no\_of\_motorcycle \* motorcycle\_toll) +(no\_of\_truck \* truck\_toll );

System.***out***.println(total\_revenue);

}

}

**class** TollBoothRevenueManagerUtil{

TollBoothRevenueManager tbr=**new** TollBoothRevenueManager();

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

**void** acceptRecord() {

System.***out***.println("enter the no of cars");

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

tbr.setNo\_of\_car(car);

System.***out***.println("enter the no of truck");

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

tbr.setNo\_of\_truck(truck);

System.***out***.println("enter the no of motorcycle");

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

tbr.setNo\_of\_motorcycle(motorcycle);

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

**float** carToll=sc.nextInt();

tbr.setCar\_toll(carToll);

System.***out***.println("toll of truck");

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

tbr.setTruck\_toll(truckToll);

System.***out***.println("toll of motorcycle");

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

tbr.setMotorcycle\_toll(motorcycleToll);

}

**void** printRecord() {

System.***out***.println("no of cars :"+ tbr.getNo\_of\_car());

System.***out***.println("no of truck :"+tbr.getNo\_of\_truck());

System.***out***.println("no of cars :"+tbr.getNo\_of\_motorcycle());

System.***out***.println("toll of cars :"+ tbr.getCar\_toll());

System.***out***.println("toll of truck :"+tbr.getTruck\_toll());

System.***out***.println("toll of cars :"+tbr.getMotorcycle\_toll());

}

**void** menuList() {

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

System.***out***.println("1.Accept Record ");

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

System.***out***.println("3.total vehicles ");

System.***out***.println("4.total revenue");

System.***out***.println("5.exit");

System.***out***.println("enter the choice");

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

**switch** (choice) {

**case** 1:

acceptRecord();

menuList();

**break**;

**case** 2:

printRecord();

menuList();

**break**;

**case** 3:

tbr.totalVehicles();

menuList();

**break**;

**case** 4:

tbr.total\_revenue();

menuList();

**break**;

**case** 5:

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

**break**;

**default**:

System.***out***.println("enter the valid choice:");

**break**;

}

}