***ASSIGNMENT -3***

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.**

**Ans- package Os\_Assignments;**

**import java.util.Scanner;**

**class LoanAmortizationCalculator {**

**private float principal;**

**private float rate;**

**private int time;**

**private float monthlyInterestRate;**

**private int numberOfMonths;**

**private float monthlyPayment;**

**public void acceptRecord() {**

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

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

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

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

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

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

**time = sc.nextInt();**

**}**

**public void calculateMonthlyPayment() {**

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

**numberOfMonths = time \* 12;**

**monthlyPayment = principal \* (monthlyInterestRate \* (1 + monthlyInterestRate)\*(numberOfMonths)\*(numberOfMonths)) / ((1 + monthlyInterestRate)\*(numberOfMonths)\*(numberOfMonths) - 1);**

**}**

**public void printRecord() {**

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

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

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

**}**

**}**

**public class Loan\_Calc {**

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

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

**LoanAmortizationCalculator l = new LoanAmortizationCalculator ();**

**l.acceptRecord();**

**l.calculateMonthlyPayment();**

**l.printRecord();**

**}**

**}**

***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.**

***Ans* – package Os\_Assignments;**

**import java.util.Scanner;**

**class CompoundInterestCalculator{**

**private double principal;**

**private double interest;**

**private int noc;**

**private int time;**

**private double futureValue;**

**private double interestEarned;**

**}**

**public void acceptRecord() {**

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

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

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

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

**interest = sc.nextDouble();**

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

**noc = sc.nextInt();**

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

**time = sc.nextInt();**

**sc.close();**

**}**

**public void calculateFutureValue() {**

**futureValue = principal \* (1 + interest / noc)\*(noc \* time)\*(noc \* time);**

**interestEarned = futureValue - principal;**

**}**

**public void printRecord(){**

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

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

**}**

**public class CI\_Calc {**

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

**CompoundInterestCalculator c = new CompoundInterestCalculator();**

**c.acceptRecord();**

**c.calculateFutureValue();**

**c.printRecord();**

***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.**

**Ans - package Os\_Assignments;**

**import java.util.Scanner;**

**class BMICalc {**

**private float weight;**

**private float height;**

**private float bmic;**

**private float Underweight;**

**private float Normalweight;**

**private float Overweight;**

**private float Obese;**

**public void acceptRecord() {**

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

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

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

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

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

**}**

**public void calculateBMI() {**

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

**}**

**public void classifyBMI () {**

**if(bmic<18.5) {**

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

**}**

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

**System.*out*.println ("Your weight is normal");**

**}**

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

**System.*out*.println ("You are overweight!");**

**}**

**else {**

**System.*out*.println ("You are OBESE!!!!");**

**}**

**}**

**public void printRecord() {**

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

**}**

**}**

**public class BMI {**

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

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

**BMICalc bm = new BMICalc();**

**bm.acceptRecord();**

**bm.calculateBMI();**

**bm.printRecord();**

**bm.classifyBMI();**

**}**

**}**

***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.

***Ans-*** package Os\_Assignments;

import java.util.Scanner;

class DiscountCalculator {

private float originalPrice;

private float discountRate ;

private float discountAmount;

private float finalPrice;

public void acceptRecord() {

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

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

originalPrice = sc.nextFloat();

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

discountRate = sc.nextFloat();

}

public void calculateDiscount () {

discountAmount = originalPrice \* (discountRate / 100);

finalPrice = originalPrice - discountAmount;

}

public void printRecord() {

System.***out***.println("₹ "+discountAmount);

System.***out***.println("₹ "+finalPrice);

}

}

public class retail {

public static void main(String[] args) {

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

DiscountCalculator dc= new DiscountCalculator();

dc.acceptRecord();

dc.calculateDiscount();

dc.printRecord();

}

}

***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.

Ans- package Os\_Assignments;

import java.util.Scanner;

class TollBoothRevenueManager {

private int cart;

private int truckt;

private int biket;

private int carb;

private int truckb;

private int bikeb;

private int total;

private int totalv;

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

public void acceptRecord() {

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

cart = sc.nextInt();

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

truckt = sc.nextInt();

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

biket = sc.nextInt();

}

public void setTollRates() {

System.***out***.println("Enter the no. of cars passing through the toll booth: ");

carb = sc.nextInt();

System.***out***.println("Enter the no. of truck passing through the toll booth: ");

truckb = sc.nextInt();

System.***out***.println("Enter the no. of bike passing through the toll booth: ");

bikeb = sc.nextInt();

}

public void calculateRevenue(){

total = (cart\*carb)+(truckt\*truckb)+(biket\*bikeb);

totalv = (carb+truckb+bikeb);

}

public void printRecord() {

System.***out***.println("Total number of vehicles passing through the toll booth: "+totalv);

System.***out***.println("Total Revenue collected: ₹ "+total);

}

}

public class Booth {

public static void main(String[] args) {

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

TollBoothRevenueManager tb = new TollBoothRevenueManager();

tb.acceptRecord();

tb.setTollRates();

tb.calculateRevenue();

tb.printRecord();

}

}