1. **LoanAmortizationCalculator**

**LoanAmortizationCalculator.java**

package in.cdac.ques1;

public class LoanAmortizationCalculator {

private double principal;

private double annualInterestRate;

private int loanTerm;

private int numberOfMonths;

private double monthlyInterestRate;

private double monthlyPayment;

private double totalAmountPaid;

LoanAmortizationCalculator(){

}

public double getPrincipal(){

return this.principal;

}

public void setPrincipal(double principal){

this.principal = principal;

}

public double getAnnualInterestRate() {

return annualInterestRate;

}

public void setAnnualInterestRate(double annualInterestRate) {

this.annualInterestRate = annualInterestRate;

}

public int getLoanTerm() {

return loanTerm;

}

public void setLoanTerm(int loanTerm) {

this.loanTerm = loanTerm;

}

public int getNumberOfMonths() {

return numberOfMonths;

}

public void setNumberOfMonths(int numberOfMonths) {

this.numberOfMonths = numberOfMonths;

}

public double getMonthlyInterestRate() {

return monthlyInterestRate;

}

public void setMonthlyInterestRate(double monthlyInterestRate) {

this.monthlyInterestRate = monthlyInterestRate;

}

public void setMonthlyPayment(double monthlyPayment) {

this.monthlyPayment = monthlyPayment;

}

public double getMonthlyPayment() {

return monthlyPayment;

}

public void setTotalAmountPaid(double totalAmountPaid){

this.totalAmountPaid = totalAmountPaid;

}

public double getTotalAmountPaid() {

return totalAmountPaid;

}

}

**LoanAmortizationCalculatorUtil.java**

package in.cdac.ques1;

import java.util.Scanner;

public class LoanAmortizationCalculatorUtil {

LoanAmortizationCalculator obj1 = new LoanAmortizationCalculator();

public void acceptRecord(Scanner sc){

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

obj1.setPrincipal( sc.nextDouble() );

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

obj1.setAnnualInterestRate( sc.nextDouble() );

System.out.println("Enter Loan Term in years: ");

obj1.setLoanTerm(sc.nextInt());

}

public void calculateMonthlyPayment(){

obj1.setNumberOfMonths(obj1.getLoanTerm() \* 12);

obj1.setMonthlyInterestRate(obj1.getAnnualInterestRate() / 12 / 100);

obj1.setMonthlyPayment(obj1.getPrincipal() \* (obj1.getMonthlyInterestRate() \* Math.pow((1 + obj1.getMonthlyInterestRate()),(obj1.getNumberOfMonths())))

/ (Math.pow((1 + obj1.getMonthlyInterestRate()),(obj1.getNumberOfMonths())) - 1));

obj1.setTotalAmountPaid(obj1.getMonthlyPayment() \* obj1.getNumberOfMonths());

}

public String toString(){

return "\nMonthly Payment: "+ String.format("%.2f",obj1.getMonthlyPayment())+

" Rs\n\nTotal Amount to be Paid: "+ String.format("%.2f",obj1.getTotalAmountPaid()) + " Rs\n\n";

}

public void printRecord(){

System.out.println(this.toString());

}

public int menuList(Scanner sc) {

System.out.println("\n1. Accept Record\n2. Print Record\n0. Exit\n");

return sc.nextInt();

}

}

**Tester.java**

package in.cdac.ques1;

import java.util.Scanner;

public class Tester {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

LoanAmortizationCalculatorUtil l1 = new LoanAmortizationCalculatorUtil();

int choice;

while ( (choice = l1.menuList(sc)) != 0) {

switch(choice) {

case 1:

l1.acceptRecord(sc);

l1.calculateMonthlyPayment();

break;

case 2: l1.printRecord();

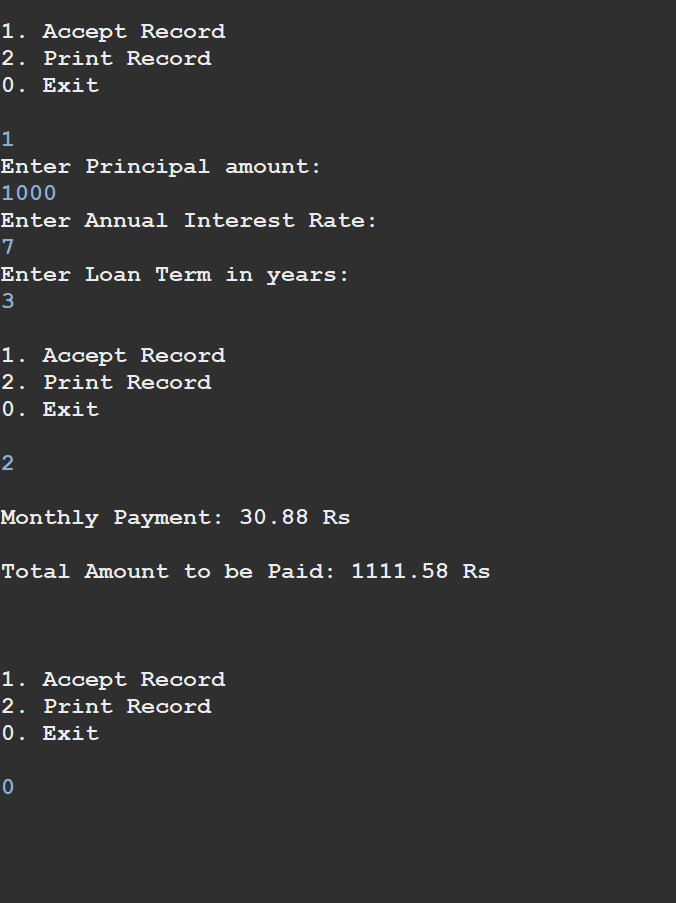
break;

}

}

}

}



1. **CompoundInterestCalculator**

**CompoundInterestCalculator.java**

package in.cdac.ques2;

public class CompoundInterestCalculator {

private double principal;

private double annualInterestRate;

private double numberOfCompounds;

private double years;

private double futureValue;

private double totalInterest;

public CompoundInterestCalculator(){

}

public double getPrincipal() {

return principal;

}

public void setPrincipal(double principal) {

this.principal = principal;

}

public double getAnnualInterestRate() {

return annualInterestRate;

}

public void setAnnualInterestRate(double annualInterestRate) {

this.annualInterestRate = annualInterestRate;

}

public double getNumberOfCompounds() {

return numberOfCompounds;

}

public void setNumberOfCompounds(double numberOfCompounds) {

this.numberOfCompounds = numberOfCompounds;

}

public double getYears() {

return years;

}

public void setYears(double years) {

this.years = years;

}

public double getFutureValue() {

return futureValue;

}

public void setFutureValue(double futureValue) {

this.futureValue = futureValue;

}

public double getTotalInterest() {

return totalInterest;

}

public void setTotalInterest(double totalInterest) {

this.totalInterest = totalInterest;

}

}

**CompoundInterestCalculatorUtil.java**

package in.cdac.ques2;

import java.util.Scanner;

public class CompoundInterestCalculatorUtil {

CompoundInterestCalculator cal = new CompoundInterestCalculator();

public void acceptRecord(Scanner sc) {

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

cal.setPrincipal(sc.nextInt());

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

cal.setAnnualInterestRate(sc.nextDouble());

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

cal.setNumberOfCompounds(sc.nextInt());

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

cal.setYears(sc.nextInt());

}

public void calculateFutureValue() {

cal.setFutureValue(cal.getPrincipal() \* Math.pow((1 + cal.getAnnualInterestRate() / cal.getNumberOfCompounds()),(cal.getNumberOfCompounds() \* cal.getYears())));

cal.setTotalInterest(cal.getFutureValue() - cal.getPrincipal());

}

public void printRecord() {

System.out.println(this.toString());

}

public String toString(){

return "\nFuture Value: "+ String.format("%.2f",cal.getFutureValue())+

" Rs\n\nTotal Interest Earned: "+ String.format("%.2f",cal.getTotalInterest()) + " Rs\n\n";

}

public int menuList(Scanner sc) {

System.out.println("\n1. Accept Record\n2. Print Record\n0. Exit\n");

return sc.nextInt();

}

}

**Program.java**

package in.cdac.ques2;

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

CompoundInterestCalculatorUtil util = new CompoundInterestCalculatorUtil();

int choice;

while ( (choice = util.menuList(sc)) != 0) {

switch(choice) {

case 1:

util.acceptRecord(sc);

util.calculateFutureValue();

break;

case 2: util.printRecord();

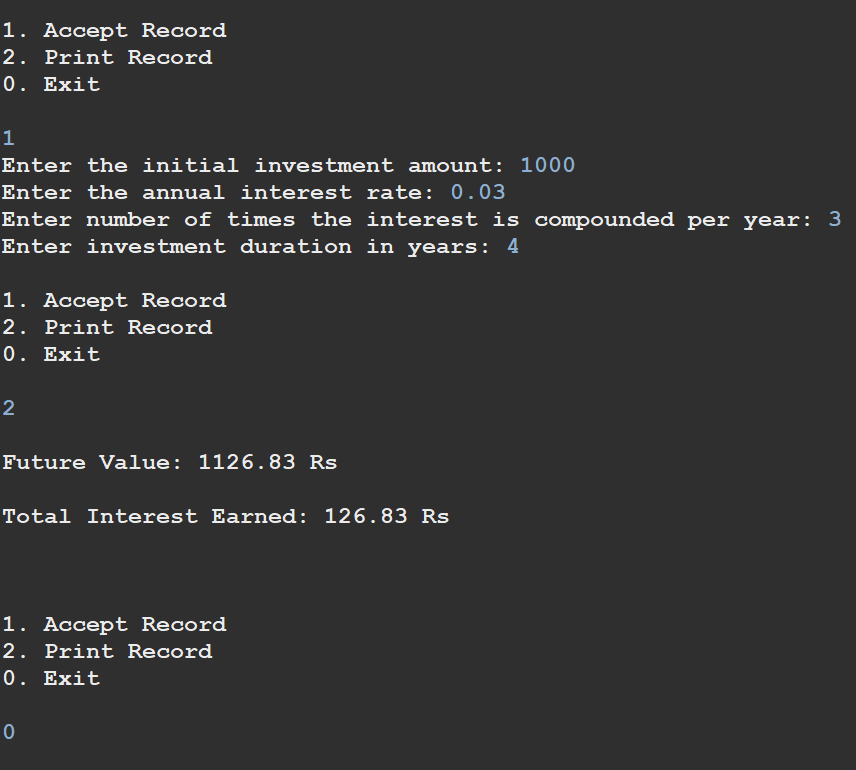
break;

}

}

}

}



1. **BMI (Body Mass Index) Tracker**

**BMITracker.java**

package in.cdac.ques3;

public class BMITracker {

private double weight;

private double height;

private double bmiValue;

private String bmiClass;

public BMITracker(){

this.bmiClass = "0";

}

public String getBmiClass() {

return bmiClass;

}

public void setBmiClass(String bmiClass) {

this.bmiClass = bmiClass;

}

public double getWeight() {

return weight;

}

public void setWeight(double weight) {

this.weight = weight;

}

public double getHeight() {

return height;

}

public void setHeight(double height) {

this.height = height;

}

public double getBmiValue() {

return bmiValue;

}

public void setBmiValue(double bmiValue) {

this.bmiValue = bmiValue;

}

}

**BMITrackerUtil.java**

package in.cdac.ques3;

import java.util.Scanner;

public class BMITrackerUtil {

BMITracker b = new BMITracker();

public void acceptRecord(Scanner sc){

System.out.print("Enter weight in kg: ");

b.setWeight(sc.nextDouble());

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

b.setHeight(sc.nextDouble());

}

public void printRecord(){

System.out.println(this.toString());

}

public void calculateBMI(){

b.setBmiValue(b.getWeight() / (b.getHeight() \* b.getHeight()));

}

public void classifyBMI(){

if(b.getBmiValue()<18.5)

b.setBmiClass("Underweight");

else if(b.getBmiValue()>=18.5 && b.getBmiValue()<25)

b.setBmiClass("Normal weight");

else if(b.getBmiValue()>=25 && b.getBmiValue()<30)

b.setBmiClass("Overweight");

else

b.setBmiClass("Obese");

}

public String toString(){

return "\nBMI: "+ String.format("%.2f",b.getBmiValue())+"\n"+b.getBmiClass();

}

public int menuList(Scanner sc) {

System.out.println("\n1. Accept Record\n2. Print Record\n0. Exit\n");

return sc.nextInt();

}

}

**Program.java**

package in.cdac.ques3;

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

BMITrackerUtil util = new BMITrackerUtil();

int choice;

while ( (choice = util.menuList(sc)) != 0) {

switch(choice) {

case 1:

util.acceptRecord(sc);

util.calculateBMI();

util.classifyBMI();

break;

case 2: util.printRecord();

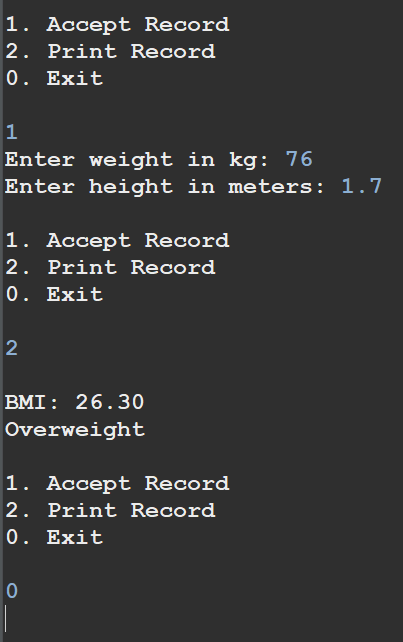
break;

}

}

}

}



1. **Discount Calculation for Retail Sales**

**DiscountCalculator.java**

package in.cdac.ques4;

public class DiscountCalculator {

private double originalPrice;

private double discountRate ;

private double discountAmount;

private double finalPrice;

public DiscountCalculator() {

}

public double getOriginalPrice() {

return originalPrice;

}

public void setOriginalPrice(double originalPrice) {

this.originalPrice = originalPrice;

}

public double getDiscountRate() {

return discountRate;

}

public void setDiscountRate(double discountRate) {

this.discountRate = discountRate;

}

public double getDiscountAmount() {

return discountAmount;

}

public void setDiscountAmount(double discountAmount) {

this.discountAmount = discountAmount;

}

public double getFinalPrice() {

return finalPrice;

}

public void setFinalPrice(double finalPrice) {

this.finalPrice = finalPrice;

}

}

**DiscountCalculatorUtil.java**

package in.cdac.ques4;

import java.util.Scanner;

public class DiscountCalculatorUtil {

DiscountCalculator dc = new DiscountCalculator();

public void acceptRecord(Scanner sc){

System.out.print("\nEnter the original price: ");

dc.setOriginalPrice(sc.nextDouble());

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

dc.setDiscountRate(sc.nextDouble());

}

public void calculateDiscount (){

dc.setDiscountAmount( dc.getOriginalPrice() \* (dc.getDiscountRate() / 100) );

dc.setFinalPrice(dc.getOriginalPrice() - dc.getDiscountAmount());

}

public void printRecord(){

System.out.println(this.toString());

}

public String toString(){

return "\nDiscount amount: "+ String.format("%.2f",dc.getDiscountAmount())+

" Rs\nFinal price: "+ String.format("%.2f",dc.getFinalPrice()) + " Rs\n\n";

}

public int menuList(Scanner sc) {

System.out.println("\n1. Accept Record\n2. Print Record\n0. Exit\n");

return sc.nextInt();

}

}

**Project.java**

package in.cdac.ques4;

import java.util.Scanner

public class Project {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

DiscountCalculatorUtil util = new DiscountCalculatorUtil();

int choice;

while ( (choice = util.menuList(sc)) != 0) {

switch(choice) {

case 1:

util.acceptRecord(sc);

util.calculateDiscount();

break;

case 2: util.printRecord();

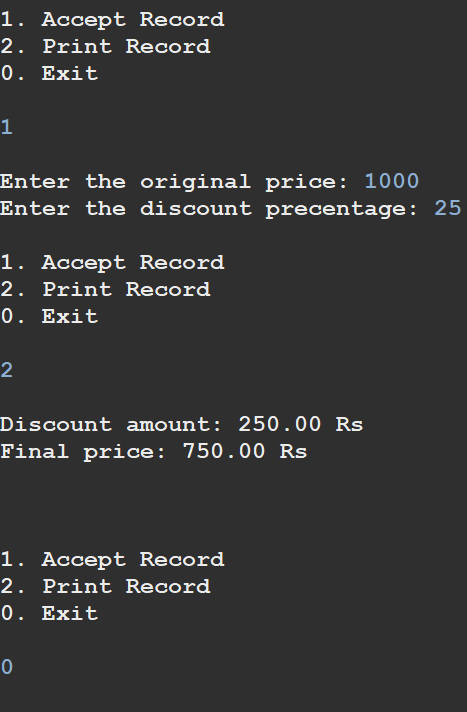
break;

}

}

}

}



1. **Toll Booth Revenue Management**

**TollBoothRevenueManager.java**

package in.cdac.ques5;

public class TollBoothRevenueManager {

private double tollRateCar;

private double tollRateTruck;

private double tollRateMotorcycle;

private int noOfCars;

private int noOfTrucks;

private int noOfMotorcycles;

private int totalNoOfVehicles;

private double totalRevenue;

public void setTollRateCar(double tollRateCar){

this.tollRateCar = tollRateCar;

}

public double getTollRateCar() {

return this.tollRateCar;

}

public double getTollRateTruck() {

return tollRateTruck;

}

public void setTollRateTruck(double tollRateTruck) {

this.tollRateTruck = tollRateTruck;

}

public double getTollRateMotorcycle() {

return tollRateMotorcycle;

}

public void setTollRateMotorcycle(double tollRateMotorcycle) {

this.tollRateMotorcycle = tollRateMotorcycle;

}

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 getNoOfMotorcycles() {

return noOfMotorcycles;

}

public void setNoOfMotorcycles(int noOfMotorcycles) {

this.noOfMotorcycles = noOfMotorcycles;

}

public int getTotalNoOfVehicles() {

return totalNoOfVehicles;

}

public void setTotalNoOfVehicles(int totalNoOfVehicles) {

this.totalNoOfVehicles = totalNoOfVehicles;

}

public double getTotalRevenue() {

return totalRevenue;

}

public void setTotalRevenue(double totalRevenue) {

this.totalRevenue = totalRevenue;

}

}

**TollBoothRevenueManagerUtil.java**

package in.cdac.ques5;

import java.util.Scanner;

public class TollBoothRevenueManagerUtil {

private TollBoothRevenueManager tm;

private static Scanner sc;

static{

sc = new Scanner(System.in);

}

TollBoothRevenueManagerUtil(){

this.tm = new TollBoothRevenueManager();

}

public void acceptRecord(){

System.out.print("\nEnter the number of Cars: ");

tm.setNoOfCars(sc.nextInt());

System.out.print("Enter the number of Trucks: ");

tm.setNoOfTrucks(sc.nextInt());

System.out.print("Enter the number of Motorcycles: ");

tm.setNoOfMotorcycles(sc.nextInt());

System.out.print("Set the toll rate for Car: ");

tm.setTollRateCar(sc.nextDouble());

System.out.print("Set the toll rate for Truck: ");

tm.setTollRateTruck(sc.nextDouble());

System.out.print("Set the toll rate for Motorcycle: ");

tm.setTollRateMotorcycle(sc.nextDouble());

}

public void calculateRevenue(){

tm.setTotalNoOfVehicles(tm.getNoOfCars() + tm.getNoOfTrucks() + tm.getNoOfMotorcycles());

tm.setTotalRevenue((tm.getTollRateCar() \* tm.getNoOfCars()) + (tm.getTollRateTruck() \* tm.getNoOfTrucks()) + (tm.getTollRateMotorcycle() \* tm.getNoOfMotorcycles()));

}

public void printRecord(){

System.out.println(this.toString());

}

public String toString(){

return "\nTotal number of vehicles: "+ String.format("%d",tm.getTotalNoOfVehicles())+

"\nTotal Revenue collected: "+ String.format("%.2f",tm.getTotalRevenue())+" Rs";

}

public static int menuList() {

System.out.println("\n1. Accept Record\n2. Print Record\n0. Exit");

System.out.print("Enter your choice: ");

return sc.nextInt();

}

}

**Program.java**

package in.cdac.ques5;

public class Program {

public static void main(String[] args) {

TollBoothRevenueManagerUtil util = new TollBoothRevenueManagerUtil();

int choice;

while ( (choice = TollBoothRevenueManagerUtil.menuList()) != 0) {

switch(choice) {

case 1:

util.acceptRecord();

util.calculateRevenue();

break;

case 2: util.printRecord();

break;

}

}

}

}

