public class Rectangle {

private int length;

private int breadth;

public void setDimensions(int len, int br) {

length = len;

breadth = br;

}

public int getLength() {

return length;

}

public int getBreadth() {

return breadth;

}

public static void main(String[] args) {

Rectangle myRectangle = new Rectangle();

myRectangle.setDimensions(5, 10);

System.out.println("Length: " + myRectangle.getLength());

System.out.println("Breadth: " + myRectangle.getBreadth());

}

}

//////////////////////////////////////////////////////

class Rectangle {

private int length;

private int breadth;

public void setDimensions(int len, int br) {

length = len;

breadth = br;

}

public void calculatePerimeter() {

int perimeter = 2 \* (length + breadth);

System.out.println("Perimeter: " + perimeter);

}

public int getLength() {

return length;

}

public int getBreadth() {

return breadth;

}

}

public class PerimeterCalculator {

public static void main(String[] args) {

Rectangle myRectangle = new Rectangle();

myRectangle.setDimensions(5, 10);

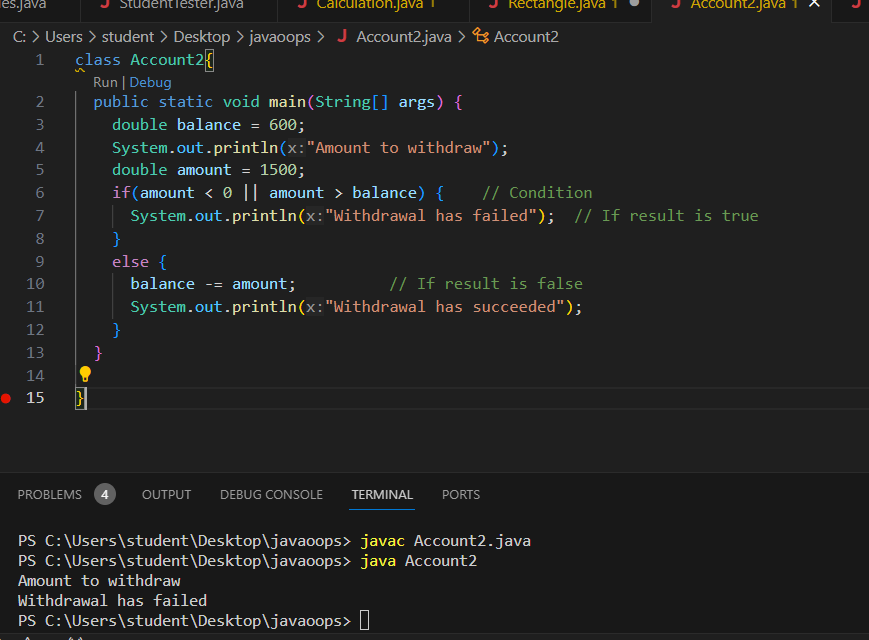
// Invoke calculatePerimeter() method on the Rectangle object

myRectangle.calculatePerimeter();

}

}

/////////////////////////////////////////////////



class Account2{

public static void main(String[] args) {

double balance = 600;

System.out.println("Amount to withdraw");

double amount = 1500;

if(amount < 0 || amount > balance) { // Condition

System.out.println("Withdrawal has failed"); // If result is true

}

else {

balance -= amount; // If result is false

System.out.println("Withdrawal has succeeded");

}

}

}

////////////////////////////////////////////////////////////////////

public class ReverseCalculator {

public static int reverseWithWhileLoop(int number) {

int reversedNumber = 0;

while (number != 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

return reversedNumber;

}

public static int reverseWithForLoop(int number) {

int reversedNumber = 0;

for (; number != 0; number /= 10) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

}

return reversedNumber;

}

public static int reverseWithDoWhileLoop(int number) {

int reversedNumber = 0;

do {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

} while (number != 0);

return reversedNumber;

}

public static void main(String[] args) {

int inputNumber = 27;

// Using while loop

System.out.println("Output(while loop):");

int reversedWhile = reverseWithWhileLoop(inputNumber);

System.out.println("ex2.1 " + reversedWhile);

// Using for loop

System.out.println("\nOutput(for loop):");

int reversedFor = reverseWithForLoop(inputNumber);

System.out.println("ex2.2 " + reversedFor);

// Using do-while loop

System.out.println("\nOutput(do-while loop):");

int reversedDoWhile = reverseWithDoWhileLoop(inputNumber);

System.out.println("ex2.3 " + reversedDoWhile);

}

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

public class FunCalculator {

public static boolean isArmstrong(int number) {

int originalNumber = number;

int sum = 0;

while (number > 0) {

int digit = number % 10;

sum += digit \* digit \* digit;

number /= 10;

}

return sum == originalNumber;

}

public static boolean isLucky(int number) {

int sum = 0;

int position = 0;

while (number > 0) {

position++;

int digit = number % 10;

if (position % 2 == 0) {

sum += digit \* digit;

}

number /= 10;

}

return sum % 9 == 0;

}

public static void main(String[] args) {

int choice = 1; // Set choice as 1 or 2

int number = 153; // Set the number for testing

if (choice == 1 || choice == 2) {

if (choice == 1 && isArmstrong(number)) {

System.out.println("ex3.1");

} else if (choice == 2 && isLucky(number)) {

System.out.println("ex3.2");

} else {

System.out.println("ex3.3");

}

} else {

System.out.println("ex3.3");

}

}

}

///////////////////////////////////////////

import java.util.Scanner;

public class FunCalculator {

public static boolean isArmstrong(int number) {

int originalNumber = number;

int sum = 0;

while (number > 0) {

int digit = number % 10;

sum += digit \* digit \* digit;

number /= 10;

}

return sum == originalNumber;

}

public static boolean isLucky(int number) {

int sum = 0;

int position = 0;

while (number > 0) {

position++;

int digit = number % 10;

if (position % 2 == 0) {

sum += digit \* digit;

}

number /= 10;

}

return sum % 9 == 0;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Calculate Armstrong");

System.out.println("Calculate lucky number");

System.out.print("Enter your choice (1 for Armstrong, 2 for lucky number): ");

int choice = scanner.nextInt();

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

int number = scanner.nextInt();

System.out.println("Your choice is: " + choice);

switch (choice) {

case 1:

if (isArmstrong(number)) {

System.out.println("The number " + number + " is an Armstrong number.");

} else {

System.out.println("The number " + number + " is not an Armstrong number.");

}

break;

case 2:

if (isLucky(number)) {

System.out.println("The number " + number + " is a lucky number.");

} else {

System.out.println("The number " + number + " is not a lucky number.");

}

break;

default:

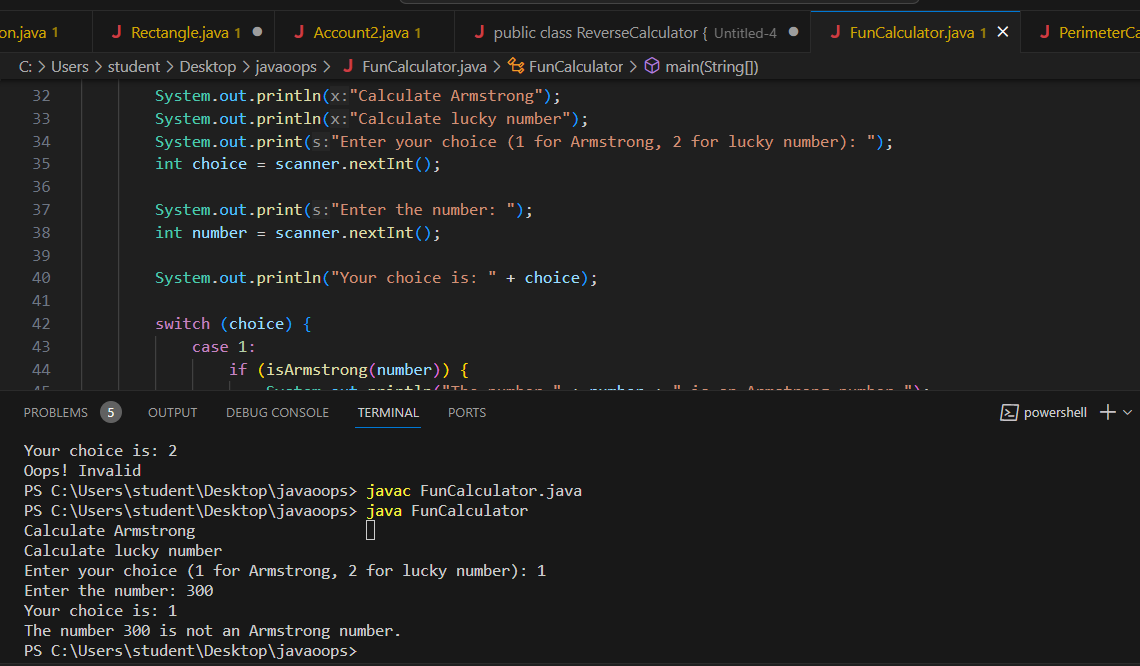
System.out.println("Oops! Invalid choice.");

}

scanner.close(); // Don't forget to close the scanner

}

}



/////////////////////////////////////////////

public class ReverseCalculator {

public static int reverseWithWhileLoop(int number) {

int reversedNumber = 0;

while (number != 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

return reversedNumber;

}

public static int reverseWithForLoop(int number) {

int reversedNumber = 0;

for (; number != 0; number /= 10) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

}

return reversedNumber;

}

public static int reverseWithDoWhileLoop(int number) {

int reversedNumber = 0;

do {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

} while (number != 0);

return reversedNumber;

}

public static void main(String[] args) {

int inputNumber = 27;

// Using while loop

int reversedWhile = reverseWithWhileLoop(inputNumber);

System.out.println("The reverse of the number using while loop is: " + reversedWhile);

// Using for loop

int reversedFor = reverseWithForLoop(inputNumber);

System.out.println("The reverse of the number using for loop is: " + reversedFor);

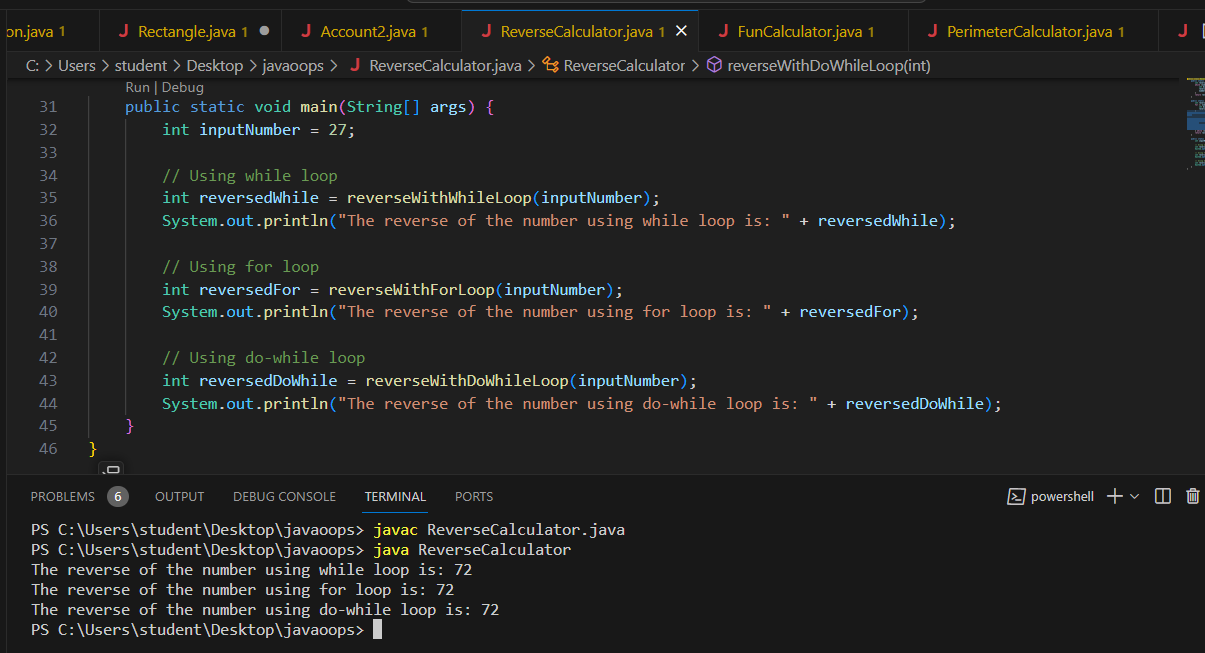
// Using do-while loop

int reversedDoWhile = reverseWithDoWhileLoop(inputNumber);

System.out.println("The reverse of the number using do-while loop is: " + reversedDoWhile);

}

}



//////////////////////////////////////////////////

class Account3{

public static void main(String[] args){

double balance = 600;

System.out.println("Amount to withdraw");

double amount = 500;

if(amount<=0){

System.out.println("Withdrawal has failed as the amount is negative");

}

else if(amount>balance){

System.out.println("Withdrawal has failed as the balance is low");

}

else{

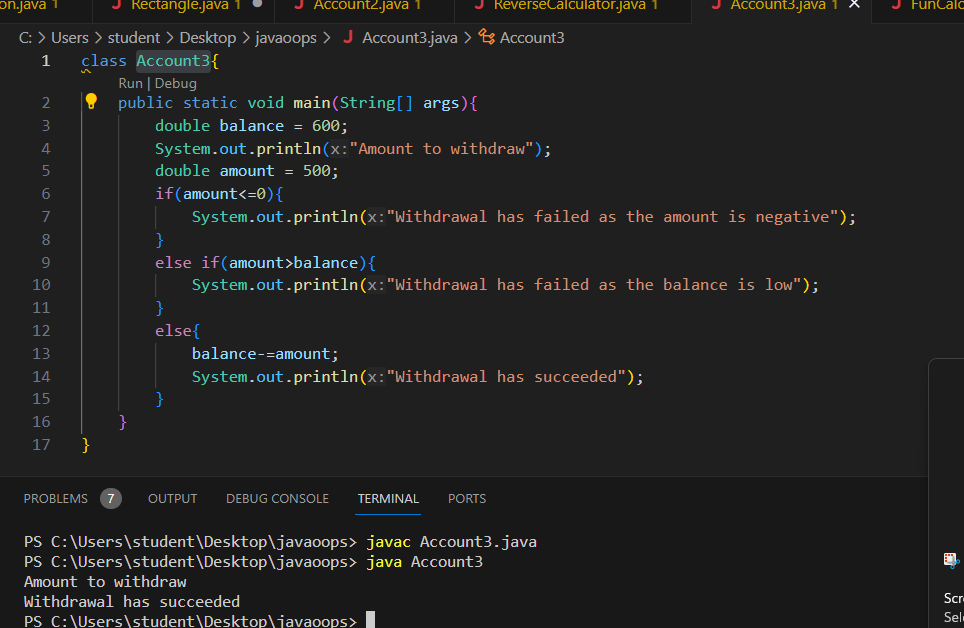
balance-=amount;

System.out.println("Withdrawal has succeeded");

}

}

}



//////////////////////////////////////////////////////////

class bank{

public static void main(String[] args){

Customer[] customer = new Customer[2]; //Reference type Array

Customer customer1 = new Customer("Anil", "Acc12345");

Customer customer2 = new Customer("Ajay", "Acc12346");

customer[0] = customer1; //storing in the array

customer[1] = customer2;

for(int i=0;i<customer.length;i++){ //traversing the array

Customer customeObject = customer[i]; //retrieving customer Object

String name = customeObject.displayCustomerName();

System.out.println("the customer name is..."+name);

}

}

}

class Customer{

private String name;

private String customerId;

Customer(String uname, String ucustomerId){

name = uname;

customerId = ucustomerId;

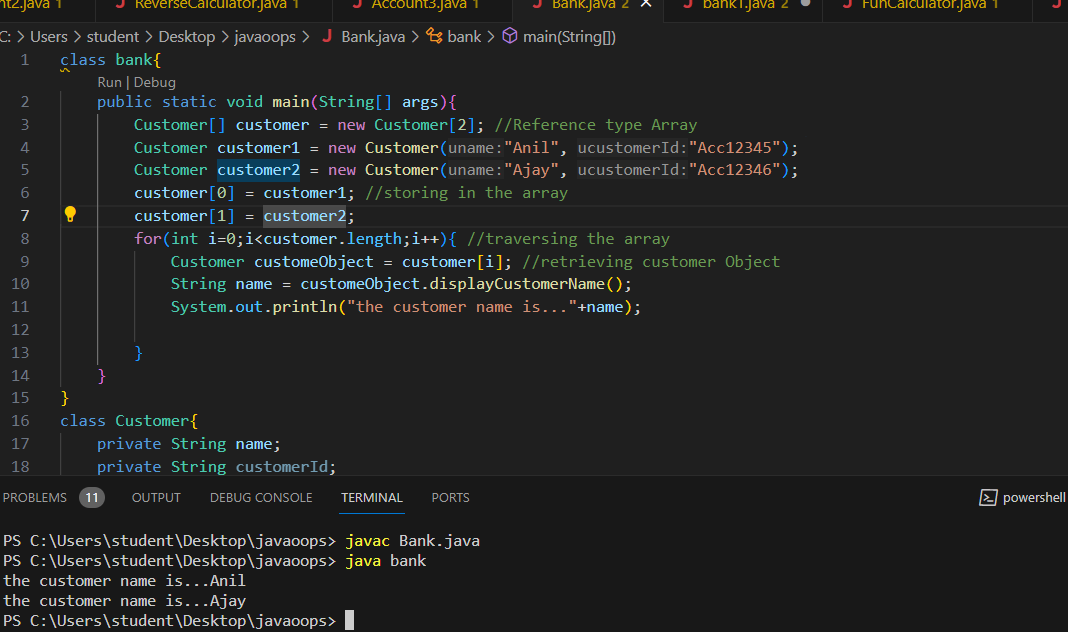
}

public String displayCustomerName(){

return name;

}

}



//////////////////////////////////////////////////////////////////////////////////////

import java.util.Scanner;

public class EmployeeRecord {

public static void main(String[] args) {

double[] salary = {23500.0, 25080.0, 28760.0, 22340.0, 19890.0};

// Calculate average salary

double totalSalary = 0;

for (double s : salary) {

totalSalary += s;

}

double averageSalary = totalSalary / salary.length;

// Count employees above and below average salary

int aboveAvgCount = 0;

int belowAvgCount = 0;

for (double s : salary) {

if (s > averageSalary) {

aboveAvgCount++;

} else if (s < averageSalary) {

belowAvgCount++;

}

}

// Display results

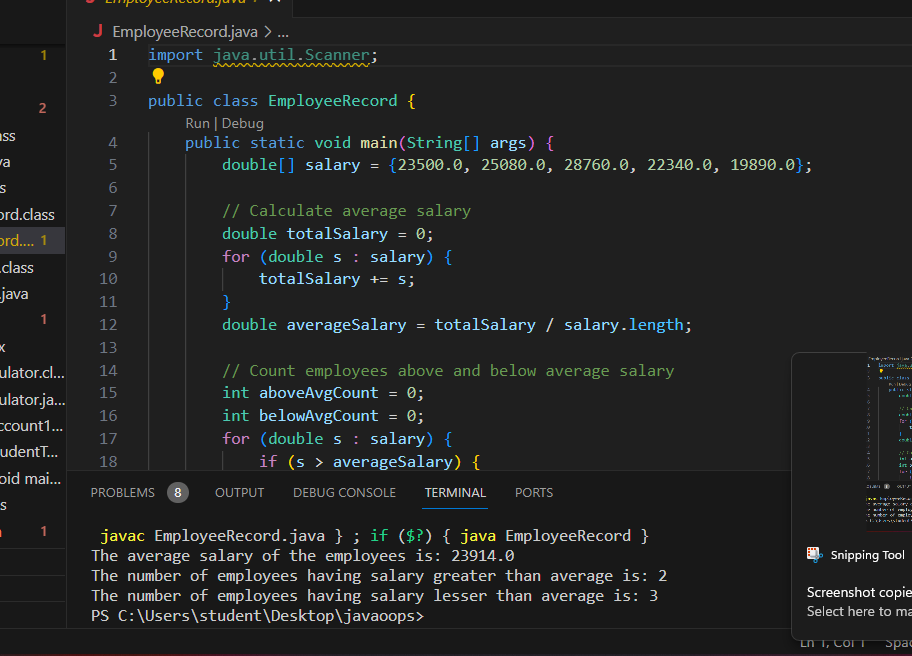
System.out.println("The average salary of the employees is: " + averageSalary);

System.out.println("The number of employees having salary greater than average is: " + aboveAvgCount);

System.out.println("The number of employees having salary lesser than average is: " + belowAvgCount);

}

}



/////////////////////////////////////////////////

class Bank {

private String bankName, area;

private String phoneNumber;

Bank() { // Default constructor

bankName = "IBank";

area = "Gandhi Nagar";

phoneNumber = "9876543210";

}

Bank(String bname, String barea, String phoneNo) { // Parameterized constructor

bankName = bname;

area = barea;

phoneNumber = phoneNo;

}

void displayBankDetails(){

System.out.println("bank Name: " + bankName);

System.out.println("Area of bank: " + area);

System.out.println("Phone number of bank: " + phoneNumber);

}

public static void main(String[] args){

Bank bank1 = new Bank(); //call default constructor

Bank bank2 = new Bank("IBank", "Jaydev Nagar", "8876543219"); //call Parameterized constructor

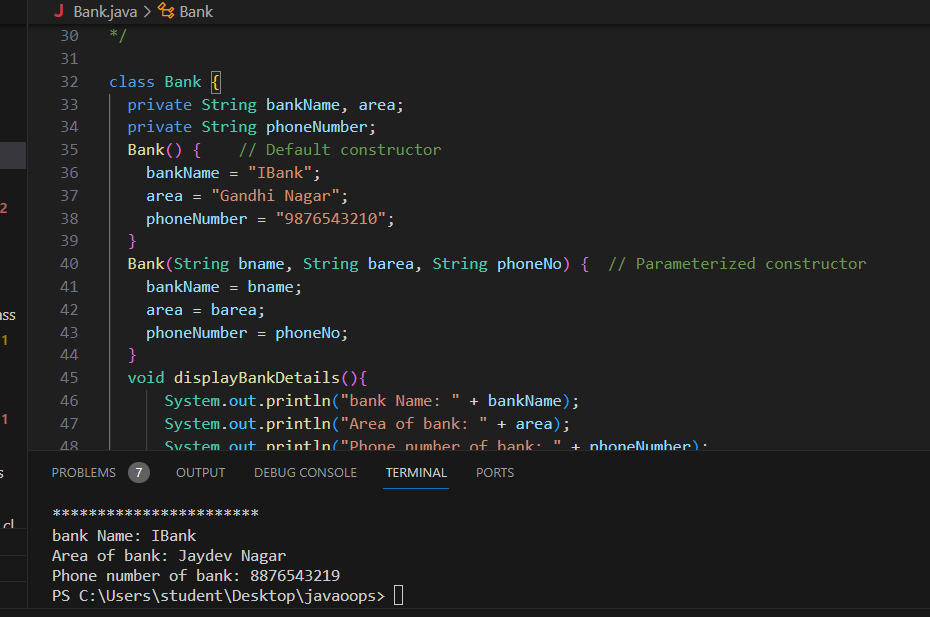
bank1.displayBankDetails();

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

bank2.displayBankDetails();

}

}



////////////////////////////////

class Bank {

private String bankName; //instance variable

private String area;

private String phoneNumber;

Bank(String bankName, String area, String phoneNumber) { // Parameterized constructor

this.bankName = bankName; //this keyword is used to assign

this.area = area; //the value for instance variables

this.phoneNumber = phoneNumber;

}

void displayBankDetails(){

System.out.println("bank Name: " + bankName);

System.out.println("Area of bank: " + area);

System.out.println("Phone number of bank: " + phoneNumber);

}

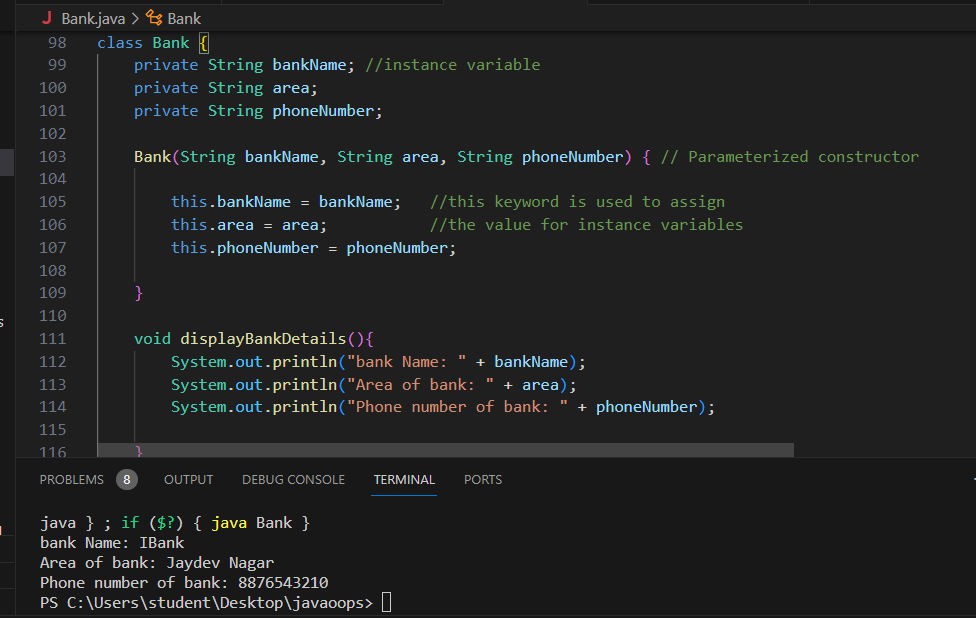
public static void main(String[] args){

Bank bank = new Bank("IBank", "Jaydev Nagar", "8876543210"); //call default constructor

bank.displayBankDetails();

}

}



//////////////////////////////////////////////////////////////////////

class Loan{

private float interest;

Loan(){

interest = 8.5f;

}

//calculateEMI overloaded methods

public double calculateEMI(int tenure, double principal){

double simpleInterest = (principal\*interest\*tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public double calculateEMI(double principal, int tenure){

double simpleInterest = (principal\*interest\*tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public double calculateEMI(int tenure, double principal, float interest){

double simpleInterest = (principal\*interest\*tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public static void main(String[] args){

Loan loan = new Loan();

double result = loan.calculateEMI(20000d,5);

double value = loan.calculateEMI(5,20000d);

double val = loan.calculateEMI(5,20000,9.5f);

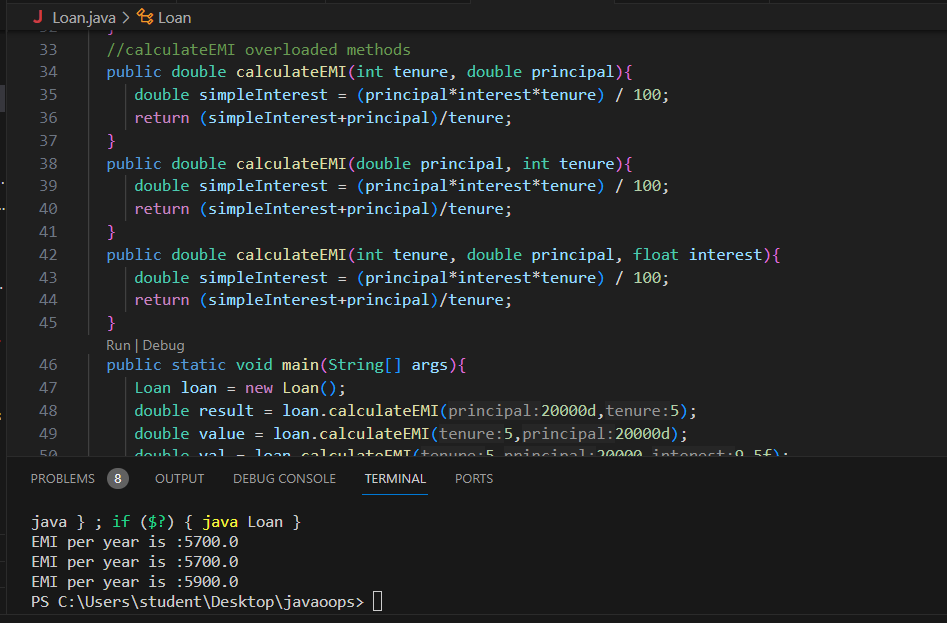
System.out.println("EMI per year is :"+result);

System.out.println("EMI per year is :"+value);

System.out.println("EMI per year is :"+val);

}

}



/////////////////////////////////////////////////////////////

class Loan{

private float interest;

Loan(){

interest = 8.5f;

}

//calculateEMI overloaded methods

public double calculateEMI(int tenure, double principal){

double simpleInterest = (principal \* interest \* tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public double calculateEMI(double principal, int tenure){

double simpleInterest = (principal \* interest \* tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public double calculateEMI(int tenure, double principal, float interest){

double simpleInterest = (principal \* interest \* tenure) / 100;

return (simpleInterest+principal)/tenure;

}

public static void main(String[] args){

Loan loan = new Loan();

double result = loan.calculateEMI(20000d, 5); //d means double

double value = loan.calculateEMI(5, 20000d);

double val = loan.calculateEMI(5, 20000, 9.5f); // f means float

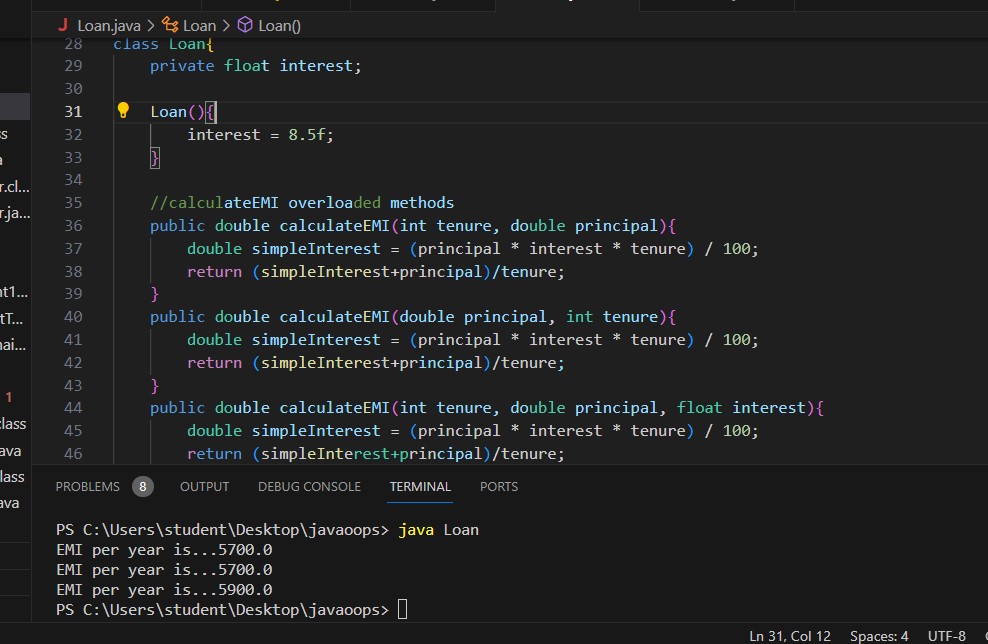
System.out.println("EMI per year is..." + result);

System.out.println("EMI per year is..." + value);

System.out.println("EMI per year is..." + val);

}

}



//////////////////////////////////////////////

class Chocolate {

    private int barCode;

    private String name;

    private int weight;

    private int cost;

    public Chocolate() {

        // Default values according to the table

        this.barCode = 101;

        this.name = "Cadbury";

        this.weight = 12;

        this.cost = 10;

    }

    // Getter and setter methods for each attribute

    public int getBarCode() {

        return barCode;

    }

    public void setBarCode(int barCode) {

        this.barCode = barCode;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getWeight() {

        return weight;

    }

    public void setWeight(int weight) {

        this.weight = weight;

    }

    public int getCost() {

        return cost;

    }

    public void setCost(int cost) {

        this.cost = cost;

    }

}

public class ChocolateTester {

    public static void main(String[] args) {

        // Create an object of Chocolate

        Chocolate chocolate = new Chocolate();

        // Use getter methods to display the values

        System.out.println("Initial Values:");

        System.out.println("BarCode: " + chocolate.getBarCode());

        System.out.println("Name: " + chocolate.getName());

        System.out.println("Weight: " + chocolate.getWeight());

        System.out.println("Cost: " + chocolate.getCost());

        // Use setter methods to modify the values

        chocolate.setBarCode(102);

        chocolate.setName("Hershey's");

        chocolate.setWeight(24);

        chocolate.setCost(50);

        // Use getter methods to display the modified values

        System.out.println("\nModified Values:");

        System.out.println("BarCode: " + chocolate.getBarCode());

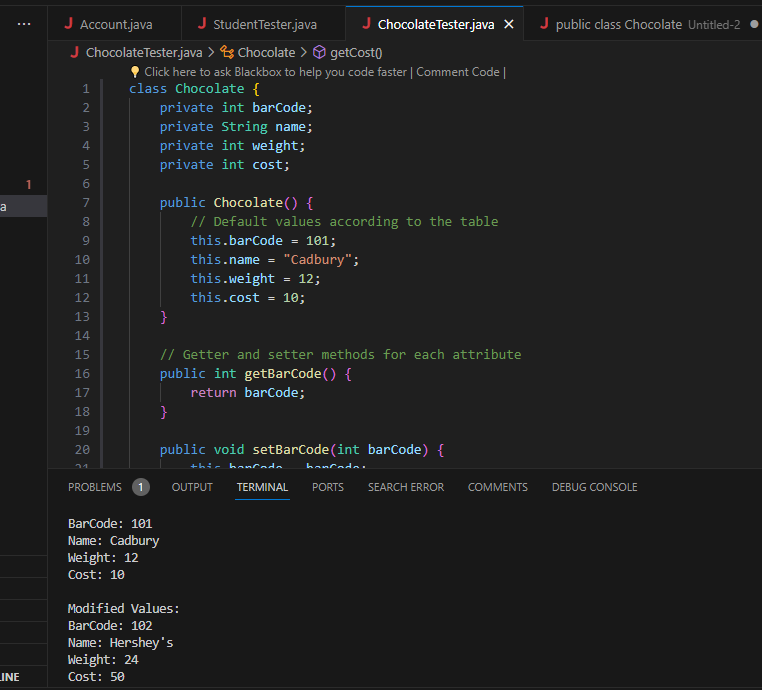
        System.out.println("Name: " + chocolate.getName());

        System.out.println("Weight: " + chocolate.getWeight());

        System.out.println("Cost: " + chocolate.getCost());

    }

}



//////////////////////////////////////////////////////////////////////

class Chocolate {

    private int barCode;

    private String name;

    private double weight;

    private double cost;

    // Default constructor

    public Chocolate() {

        this(101, "Cadbury", 12, 10); // Calling parameterized constructor with default values

    }

    // Parameterized constructor using 'this' keyword

    public Chocolate(int barCode, String name, double weight, double cost) {

        this.barCode = barCode;

        this.name = name;

        this.weight = weight;

        this.cost = cost;

    }

    // Getter and setter methods for each attribute

    public int getBarCode() {

        return barCode;

    }

    public void setBarCode(int barCode) {

        this.barCode = barCode;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public double getWeight() {

        return weight;

    }

    public void setWeight(double weight) {

        this.weight = weight;

    }

    public double getCost() {

        return cost;

    }

    public void setCost(double cost) {

        this.cost = cost;

    }

}

public class ChocolateTester {

    public static void main(String[] args) {

        // Create an object of Chocolate using default constructor

        Chocolate chocolate = new Chocolate();

        // Display initial values using getter methods

        System.out.println("Initial Values:");

        System.out.println("BarCode: " + chocolate.getBarCode());

        System.out.println("Name: " + chocolate.getName());

        System.out.println("Weight: " + chocolate.getWeight());

        System.out.println("Cost: " + chocolate.getCost());

        // Modify values using setter methods

        chocolate.setBarCode(102);

        chocolate.setName("Hershey's");

        chocolate.setWeight(24);

        chocolate.setCost(50);

        // Display modified values using getter methods

        System.out.println("\nModified Values:");

        System.out.println("BarCode: " + chocolate.getBarCode());

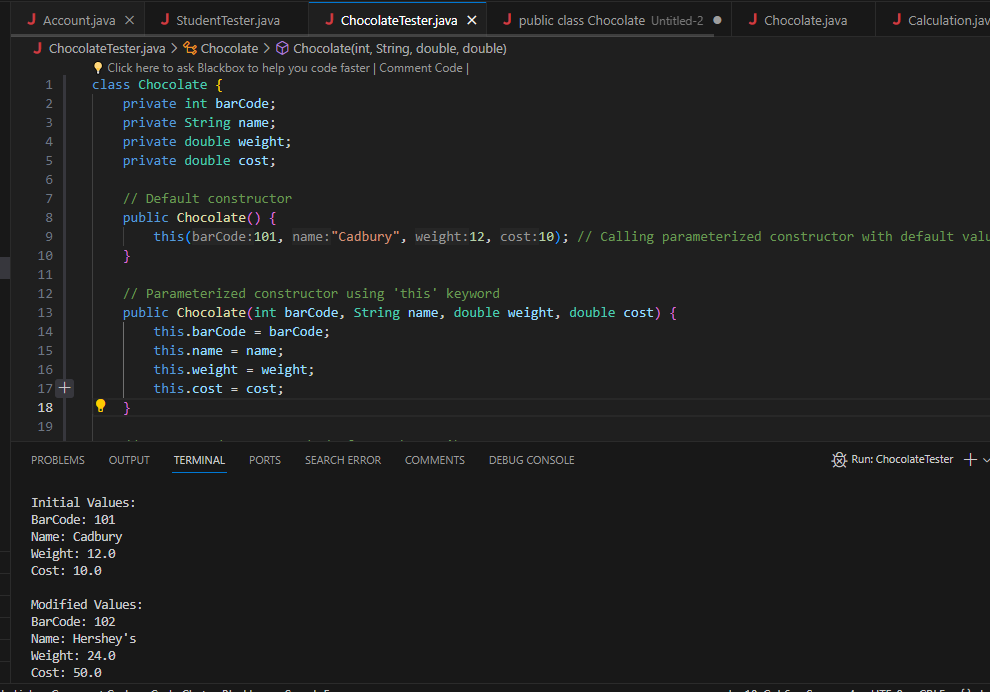
        System.out.println("Name: " + chocolate.getName());

        System.out.println("Weight: " + chocolate.getWeight());

        System.out.println("Cost: " + chocolate.getCost());

    }

}



//////////////////////////////////////////////////////////////

class Employee{

int empId;

String name;

double salary;

public double getSalary()

{

return salary;

}

public void setSalary(double salary)

{

this.salary=salary;

}

public int getEmpId()

{

return empId;

}

public void setEmpId(int empId)

{

this.empId=empId;

}

public String getName()

{

return name;

}

public void setName(String name)

{

this.name=name;

}

}

class PermanentEmployee extends Employee{

double basicPay,hra;

int experience;

public double getBasicPay(){

return basicPay;

}

public void setBasicPay(double basicPay){

this.basicPay=basicPay;

}

public double getHra(){

return hra;

}

public void setHra(double hra){

this.hra=hra;

}

public int getExperience(){

return experience;

}

public void setExperience(int experience){

this.experience=experience;

}

public void calculateSalary(){

double varCom;

if(experience<3){

varCom=0;

}

else if((experience>=3) && (experience<5)){

varCom=basicPay\*0.05;

}

else if((experience>=5) && (experience>7)){

varCom=basicPay\*0.07;

}

else{

varCom=basicPay\*0.12;

}

salary=varCom+basicPay+hra;

System.out.println("Permanent Employee : Your salary is: "+salary);

}

}

class ContractEmployee extends Employee{

double wages;

int hours;

public double getWages(){

return wages;

}

public void setWages(double wages){

this.wages=wages;

}

public int getHours(){

return hours;

}

public void setHours(int hours){

this.hours=hours;

}

public void calculateSalary(){

System.out.println("Contract Employee: Your salary is: "+(hours\*wages));

}

}

class EmployeeRecords{

public static void main(String args[]){

PermanentEmployee pemp=new PermanentEmployee();

pemp.setName("Anil");

pemp.setEmpId(101);

pemp.setBasicPay(10000);

pemp.setHra(1500);

pemp.setExperience(3);

pemp.calculateSalary();

ContractEmployee cemp=new ContractEmployee();

cemp.setName("Ankit");

cemp.setEmpId(102);

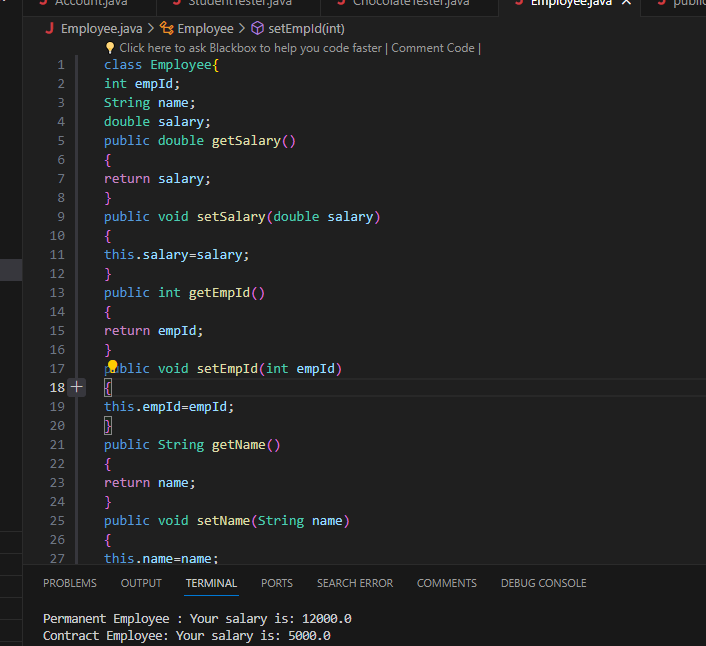
cemp.setWages(500);

cemp.setHours(10);

cemp.calculateSalary();

}

}



///////////////////////////////////////////////////////

class PlayerRating {

    private int playerPosition;

    private String playerName;

    private float criticOneRating;

    private float criticTwoRating;

    private float criticThreeRating;

    private float averageRating;

    private char category;

    public PlayerRating(int playerPosition, String playerName) {

        this.playerPosition = playerPosition;

        this.playerName = playerName;

    }

    public void calculateAverageRating(float criticOneRating, float criticTwoRating) {

        this.averageRating = (criticOneRating + criticTwoRating) / 2;

    }

    public void calculateAverageRating(float criticOneRating, float criticTwoRating, float criticThreeRating) {

        this.averageRating = (criticOneRating + criticTwoRating + criticThreeRating) / 3;

    }

    public void calculateCategory() {

        if (averageRating > 8) {

            category = 'A';

        } else if (averageRating > 5 && averageRating <= 8) {

            category = 'B';

        } else {

            category = 'C';

        }

    }

    public void display() {

        System.out.println("the player name is " + playerName);

        System.out.println("the player position is " + playerPosition);

        System.out.println("the average rating is " + averageRating);

        System.out.println("the category is " + category);

    }

}

public class PlayerRatingTester {

    public static void main(String[] args) {

        PlayerRating player1 = new PlayerRating(1, "Beckham");

        player1.calculateAverageRating(9, 9.9f);

        player1.calculateCategory();

        player1.display();

        PlayerRating player2 = new PlayerRating(1, "Oscar");

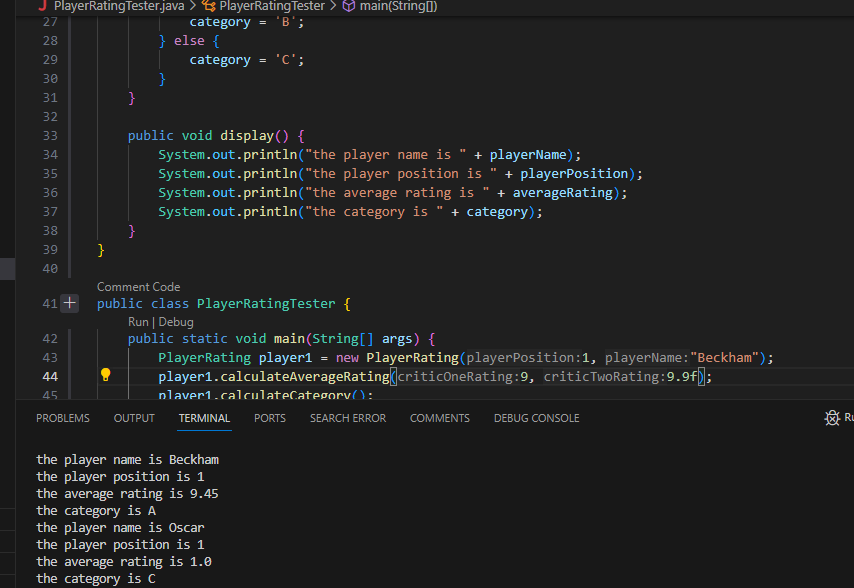
        player2.calculateAverageRating(1, 1, 1);

        player2.calculateCategory();

        player2.display();

    }

}



/////////////////////////////////////////////////////////////////

public class Registration {

public String customerName;

public String panCardNo;

public int voterId;

public String passportNo;

public int licenseNo;

public long[] telephoneNo;

public Registration(String customerName, String passportNo, long[] telephoneNo) {

this.customerName = customerName;

this.passportNo = passportNo;

this.telephoneNo = telephoneNo;

}

public Registration(String customerName, int licenseNo, String panCardNo, long[] telephoneNo) {

this.customerName = customerName;

this.panCardNo = panCardNo;

this.licenseNo = licenseNo;

this.telephoneNo = telephoneNo;

}

public Registration(String customerName, int voterId, int licenseNo, long[] telephoneNo) {

this.customerName = customerName;

this.voterId = voterId;

this.licenseNo = licenseNo;

this.telephoneNo = telephoneNo;

}

public Registration(String customerName, String panCardNo, int voterId, long[] telephoneNo) {

this.customerName = customerName;

this.panCardNo = panCardNo;

this.voterId = voterId;

this.telephoneNo = telephoneNo;

}

public String getCustomerName() {

return customerName;

}

public String getPanCardNo() {

return panCardNo;

}

public int getVoterId() {

return voterId;

}

public String getPassportNo() {

return passportNo;

}

public int getLicenseNo() {

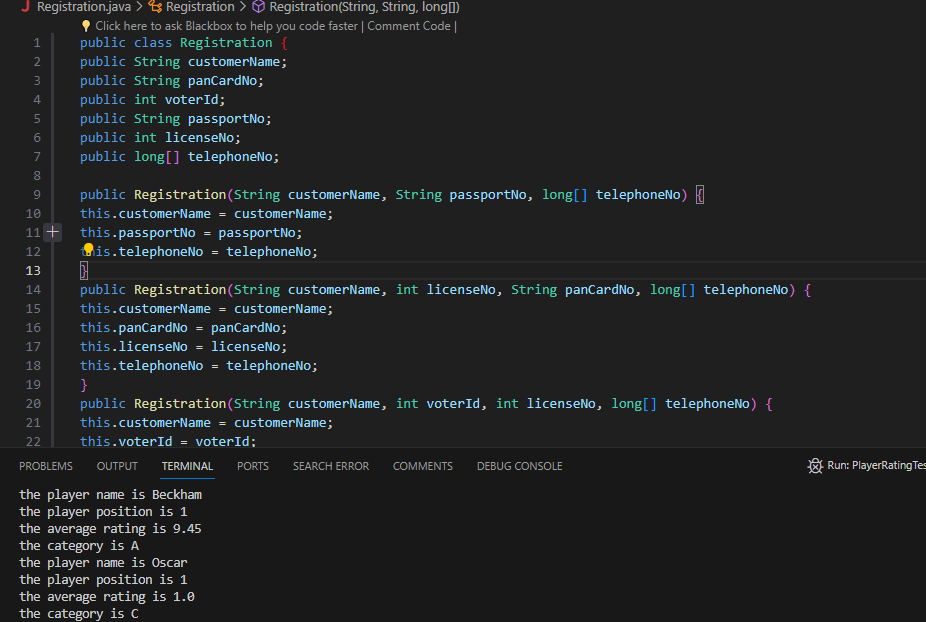
return licenseNo;

}

public long[] getTelephoneNo() {

return telephoneNo;

}



///////////////////////////////////////////////////////

class Loan{

public double calculateEMI(double principal){

double simpleInterest = (principal \* 8.5 \* 5) / 100;

return (simpleInterest+principal)/5;

}

}

class HomeLoan extends Loan {

// method overriden

public double calculateEMI(double principal){

int additionalTax = 200;

double simpleInterest = (principal \* 7.5 \* 20) / 100;

double emi = (simpleInterest + principal) / 20;

return emi+additionalTax;

}

}

class VehicleLoan extends Loan {

// method overriden

public double calculateEMI(double principal){

int additionalTax = 200;

double simpleInterest = (principal \* 9.5 \* 10) / 100;

double emi = (simpleInterest + principal) / 10;

return emi+additionalTax;

}

}

class ExecuteLoan{

public static void main(String[] args){

Loan loan = null;

loan = new HomeLoan(); //Runtime Polymorphism

double hloan = loan.calculateEMI(2000000);

loan = new VehicleLoan(); //sup class reference holding sub class Object

double vloan = loan.calculateEMI(100000);

System.out.println("Home loan emi per year is..." + hloan);

System.out.println("Vehicle loan emi per year is..." + vloan);

}

}

