**Periodic Assessment -1**

**PROGRAM-1**

**Write an interactive java program to compute the total wages based on the number of hours worked. The wages are calculated at a rate of 8.25 per hour for hours less than 40 and at the rate of 1.5 for any hours greater than 40. Capture the personal information of 3 labourers and display their wages along with the details captured. For example, if the person worked for 45 hours the wages should be (40\*8.25)+(5\*1.5).**

import java.util.\*;

public class MainLabour {

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        Labour []lbs = new Labour[3];

        for(int i = 0; i < 3; i++){

            String n; int a, hr;

            System.out.print("Enter name of labour "+(i+1)+" :");

            n = sc.nextLine();

            System.out.print("\nEnter age of labour "+(i+1)+" :");

            a = sc.nextInt();

            System.out.print("\nEnter working hours of labour "+(i+1)+" :");

            hr = sc.nextInt();

            sc.nextLine();

            System.out.print("\n----------------------------------------------------\n");

            lbs[i] = new Labour(n, a, hr);

        }

        for(int i = 0; i < 3; i++){

            lbs[i].display();

        }

    }

}

class Labour{

    String name;

    int age;

    int workingHours;

    float wage;

    Labour(String n, int a, int hr){

        name = n;

        age = a;

        workingHours = hr;

        wages();

    }

    private void wages(){

        if(workingHours > 40){

            wage = (40\*8.25f) + ((workingHours-40)\*1.5f);

        }else{

            wage = workingHours\*8.25f;

        }

    }

    public void display(){

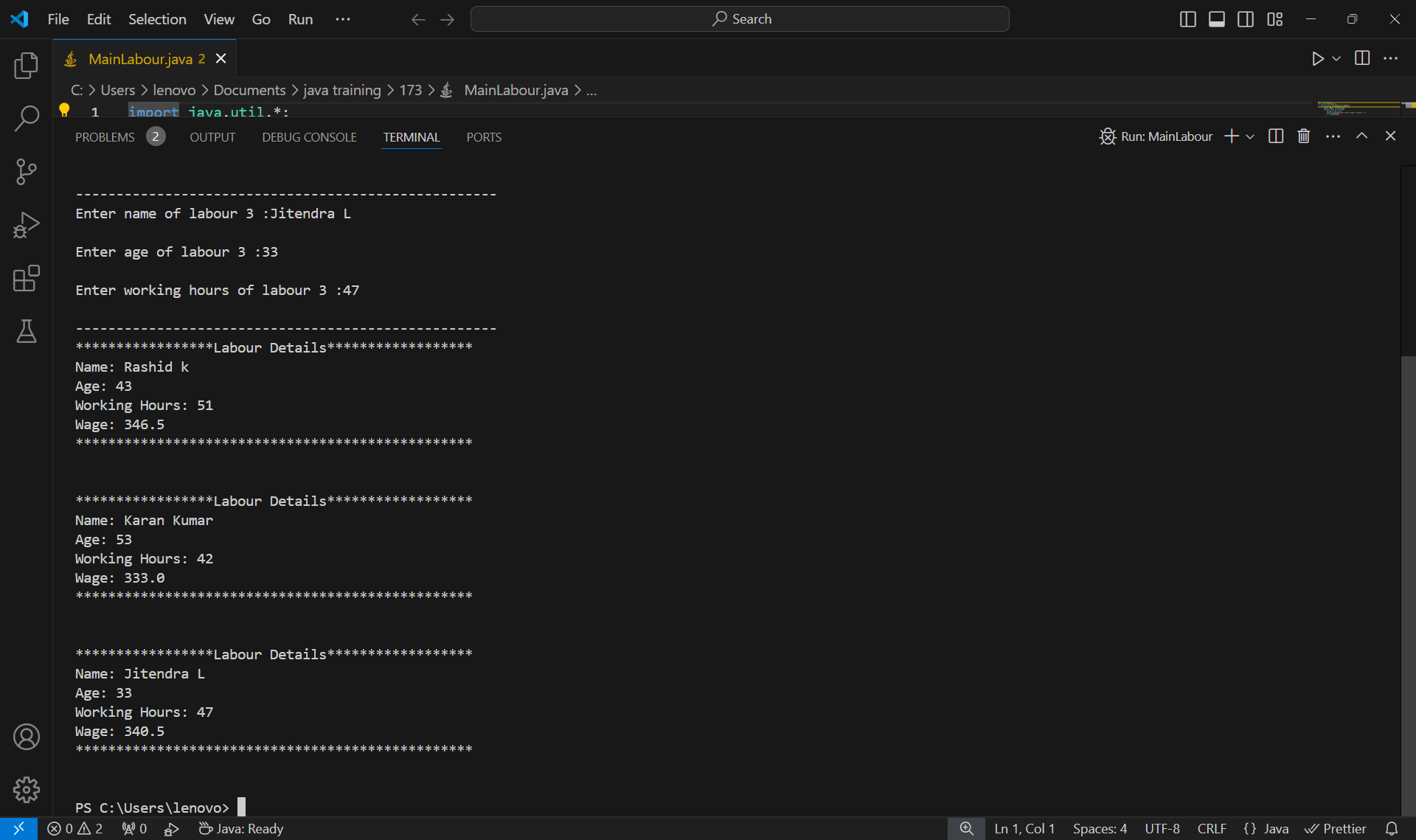
        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Labour Details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

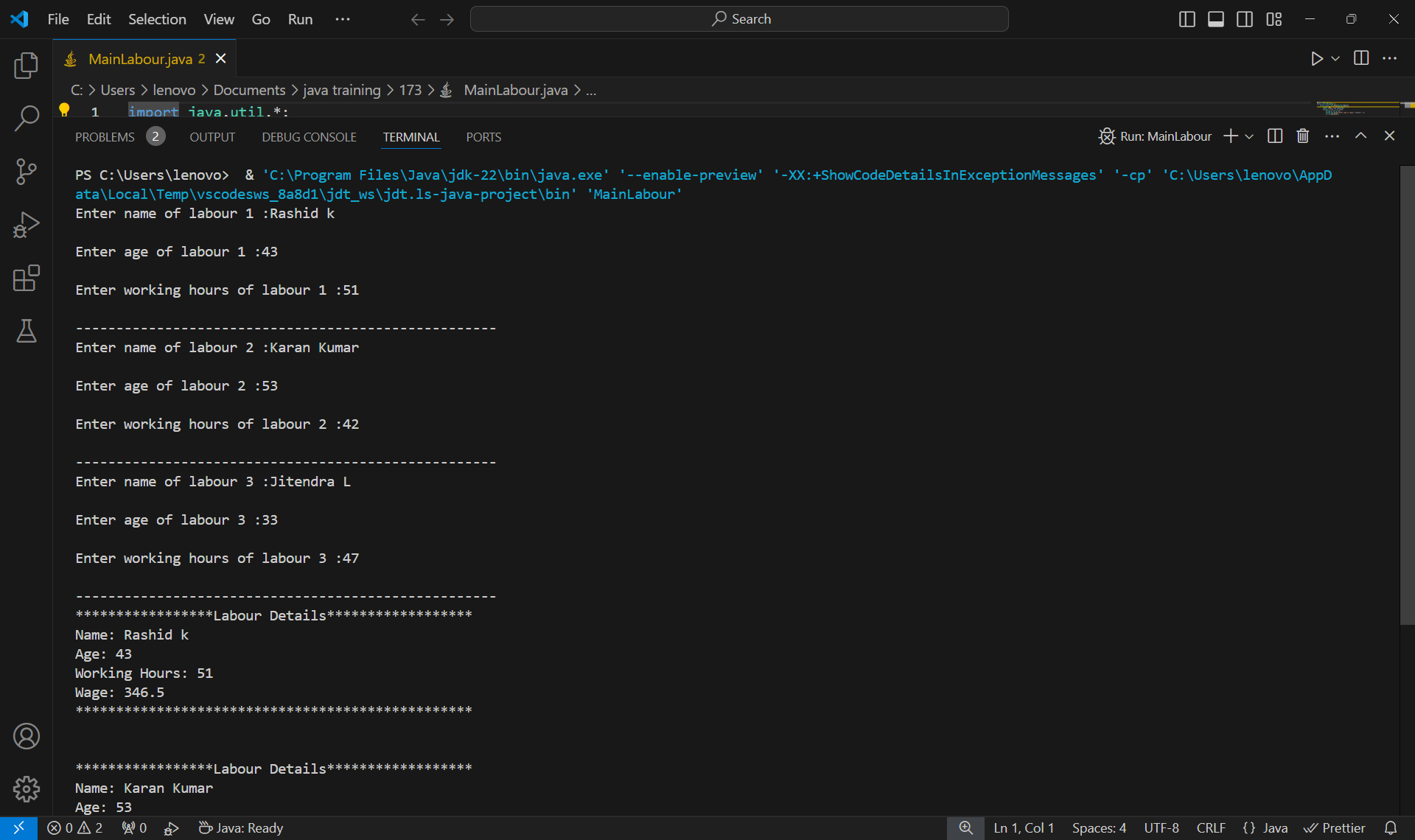
        System.out.println("Name: "+name+"\nAge: "+age+"\nWorking Hours: "+workingHours+"\nWage: "+wage);

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

    }

}





**PROGRAM-2**

**Write a Java program to compute the reverse of a number and check whether the reversed number is prime or not. Capture the user input through Scanner class**

import java.util.Scanner;

public class MainPrime {

    public static void main(String[] args) {

        isPrime();

    }

    public static void isPrime(){

        int num, rev=0;

        Scanner sc = new Scanner(System.in);

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

        num = sc.nextInt();

        while(num > 0){

            rev = rev\*10 + num%10;

            num /= 10;

        }

        if (rev <= 1) {

            System.out.println("Reversed number "+rev+" is not prime");

            return;

        }

        for (int i = 2; i <= Math.sqrt(rev); i++) {

            if (rev % i == 0) {

                System.out.println("Reversed number "+rev+" is not prime");

                return;

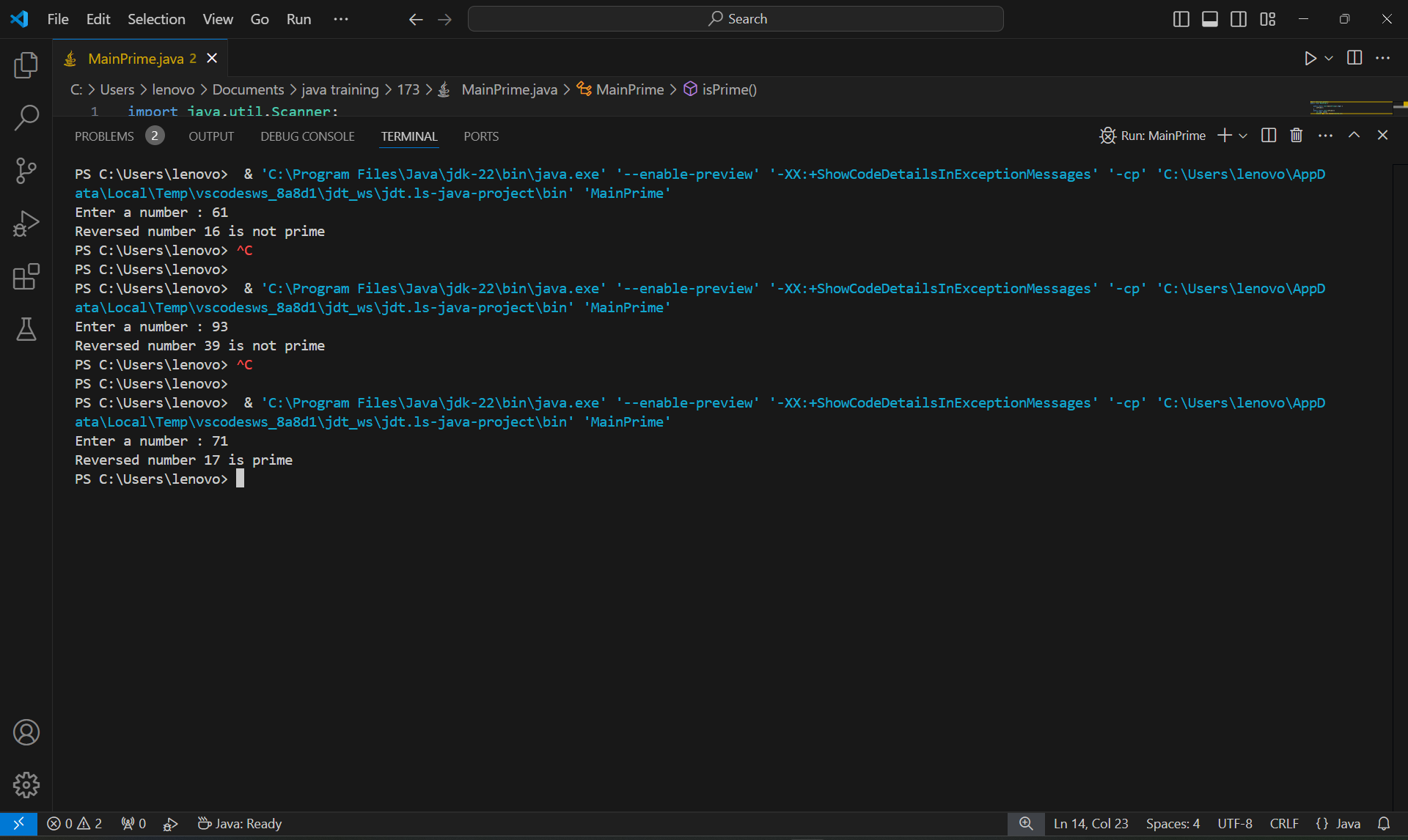
            }

        }

        System.out.println("Reversed number "+rev+" is prime");

    }

}



**PROGRAM-3**

**Write a program to capture the name, age, gender, qualification, salary of five different people and display number of persons whose age is greater than 40**

import java.util.Scanner;

public class MainAge {

    public static void main(String[] args) {

        Person[] persons = new Person[5];

        for (int i = 0; i < 5; i++) {

            persons[i] = new Person();

            persons[i].inputDetails();

        }

        int countAbove40 = 0;

        for (Person person : persons) {

            if (person.years > 40) {

                countAbove40++;

            }

        }

        System.out.println("Number of people above 40: " + countAbove40);

    }

}

class Person {

    public String fullName;

    public int years;

    public char sex;

    public String eduLevel;

    public float income;

    public void displayDetails() {

        System.out.println("Name: " + fullName);

        System.out.print("Age (in years): " + years);

        System.out.print("Gender: " + sex);

        System.out.print("Qualification: " + eduLevel);

        System.out.print("Salary: " + income);

        System.out.println();

    }

    public void inputDetails() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter name: ");

        fullName = scanner.next();

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

        years = scanner.nextInt();

        System.out.print("Enter gender: ");

        sex = scanner.next().charAt(0);

        System.out.print("Enter qualification: ");

        scanner.nextLine(); // Consume the leftover newline

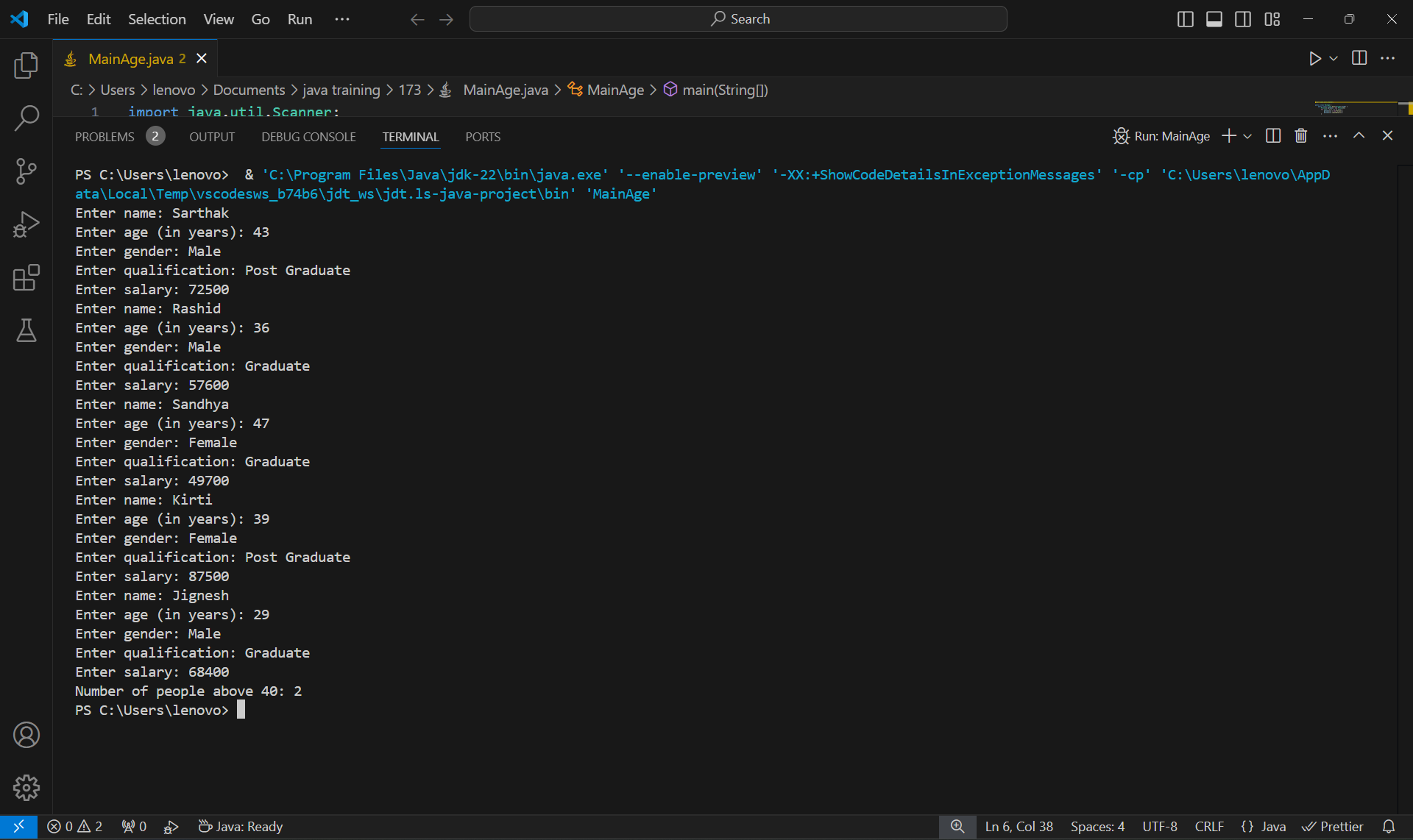
        eduLevel = scanner.nextLine();

        System.out.print("Enter salary: ");

        income = scanner.nextFloat();

    }

}



**PROGRAM-4**

**A small airline has just purchased the computer for its new automated reservations system. You have been asked to program the new system in Java to assign seats on each flight of the airline’s two planes, each of capacity: 10.**

**Define a user defined class to represent the reservation details like passenger name, mobile number, flight number and reserved seat number.**

**Keep the flight details in two static String arrays for each flight. The first five seats (index 0 to 4) represent the First Class whereas the next five seats (index 5 to 9) represent the Economy Class. Initially, both the arrays should be assigned with the value Available through static block so, no booking has done. It should be updated as Reserved for each corresponding booking.**

**Define a static method to display the flight details. Sample is here:**

**• Define a constructor with the parameters passenger name, mobile number**

**• Create a static method booking for every reservation. It should get the flight number and travel class (First or Economy) as parameters. If the seat is available in the corresponding flight it should return the seat number, otherwise -1. Also, the status of the corresponding flight seat should be updated as “Reserved” when it is available.**

**• Create a non-static method to display the reservation details.**

**Create a demo class which contains main method. Declare array of objects with the size 20 to store the reservation details. Create a menu driven loop to do the following with the choices from 1 to 3.**

**1. Display Reserved Passenger Details**

**2. Reserve a seat**

**3. Stop**

**The flight details should be displayed when the user press 1. The reservation details should be displayed when the user press 2. If the user press 3, the system should get the flight number and travel class as input. Then it should check the availability of the seat. If it is available, then the system collects the user name and mobile number. Now, it should create an object belonging to reservation class with complete details. Suppose the seat is not available, print the message “Next Flight leaves in 3 hours”.**

**Stop this iteration when user press 4. Display ‘choice is wrong, try again’ when user didn’t press the correct choice.**

import java.util.Scanner;

class Passenger {

    String name, mobile;

    int flightNumber, seatNumber;

    static String[] flight1 = new String[10];

    static String[] flight2 = new String[10];

    Passenger(String name, String mobile, int flightNumber, int seatNumber) {

        this.name = name;

        this.mobile = mobile;

        this.flightNumber = flightNumber;

        this.seatNumber = seatNumber;

    }

    static {

        for (int i = 0; i < 10; i++) {

            flight1[i] = "Available";

            flight2[i] = "Available";

        }

    }

    static int booking(int flightNumber, String travelClass) {

        String[] flight = (flightNumber == 1) ? flight1 : flight2;

        int start = (travelClass.equalsIgnoreCase("First")) ? 0 : 5;

        int end = start + 5;

        for (int i = start; i < end; i++) {

            if (flight[i].equals("Available")) {

                flight[i] = "Reserved";

                return i + 1;

            }

        }

        return -1;

    }

    void display() {

        System.out.println("\nPassenger Details\nName: " + name + "\nMobile: " + mobile + "\nFlight Number: " + flightNumber + "\nSeat Number: " + seatNumber);

    }

}

public class DEMO24MCA0173 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Passenger[] reservations = new Passenger[20];

        int reservationCount = 0;

        while (true) {

            System.out.println("\nMenu:\n1. Display Reserved Passenger Details\n2. Reserve a Seat\n3. Stop");

            int choice = sc.nextInt();

            switch (choice) {

                case 1:

                    if (reservationCount == 0) {

                        System.out.println("No reservations yet.");

                    } else {

                        for (int i = 0; i < reservationCount; i++) {

                            reservations[i].display();

                        }

                    }

                    break;

                case 2:

                    System.out.println("Enter flight number (1 or 2):");

                    int flightNumber = sc.nextInt();

                    sc.nextLine();

                    System.out.println("Enter travel class (First or Economy):");

                    String travelClass = sc.nextLine();

                    int seatNumber = Passenger.booking(flightNumber, travelClass);

                    if (seatNumber != -1) {

                        System.out.println("Enter passenger name:");

                        String name = sc.nextLine();

                        System.out.println("Enter mobile number:");

                        String mobile = sc.nextLine();

                        reservations[reservationCount++] = new Passenger(name, mobile, flightNumber, seatNumber);

                        System.out.println("Seat reserved successfully. Seat number: " + seatNumber+"\n\n\n");

                    } else {

                        System.out.println("Next Flight leaves in 3 hours");

                    }

                    break;

                case 3:

                    System.out.println("Stopping the system. Thank You!");

                    sc.close();

                    return;

                default:

                    System.out.println("Choice is wrong, try again");

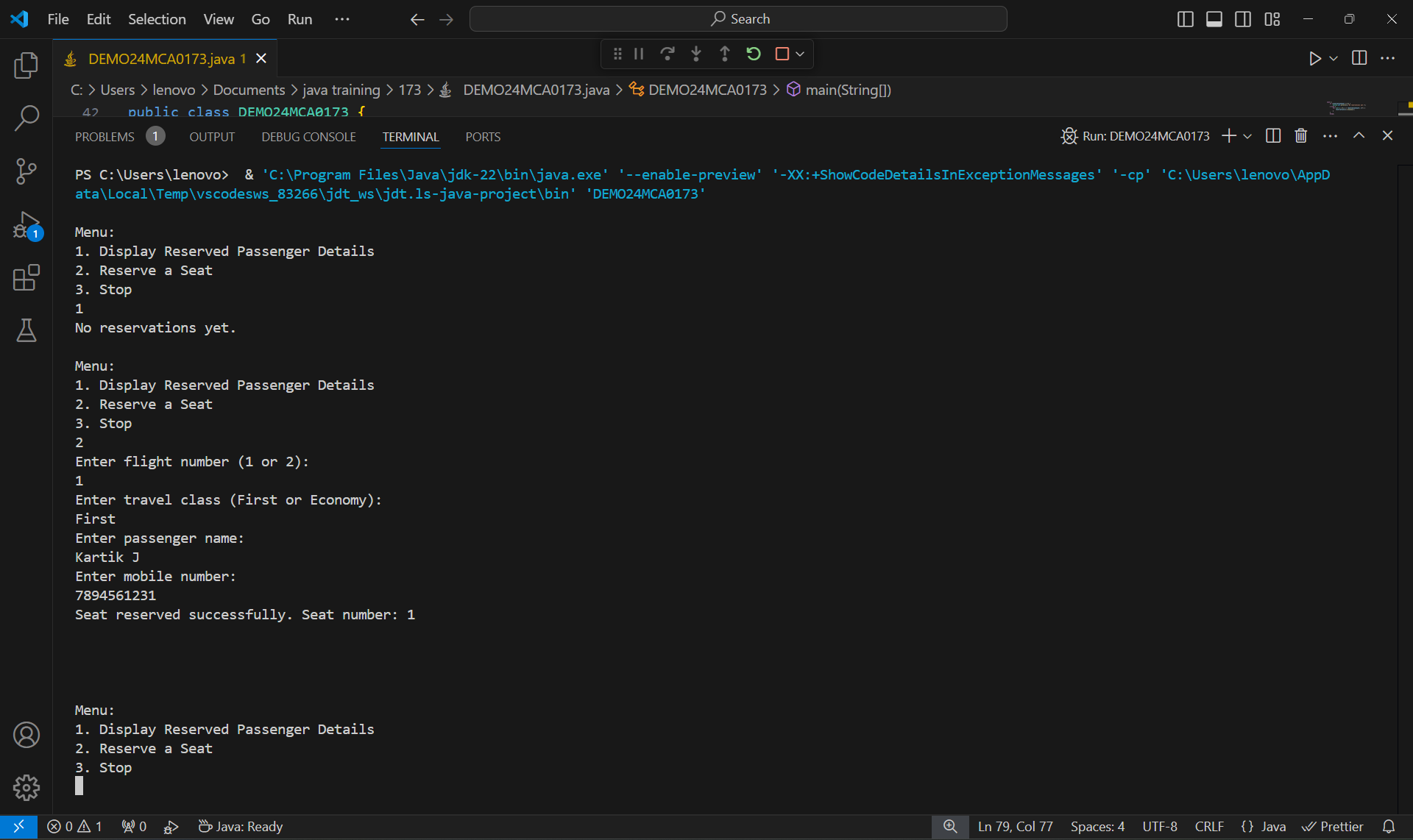
                    break;

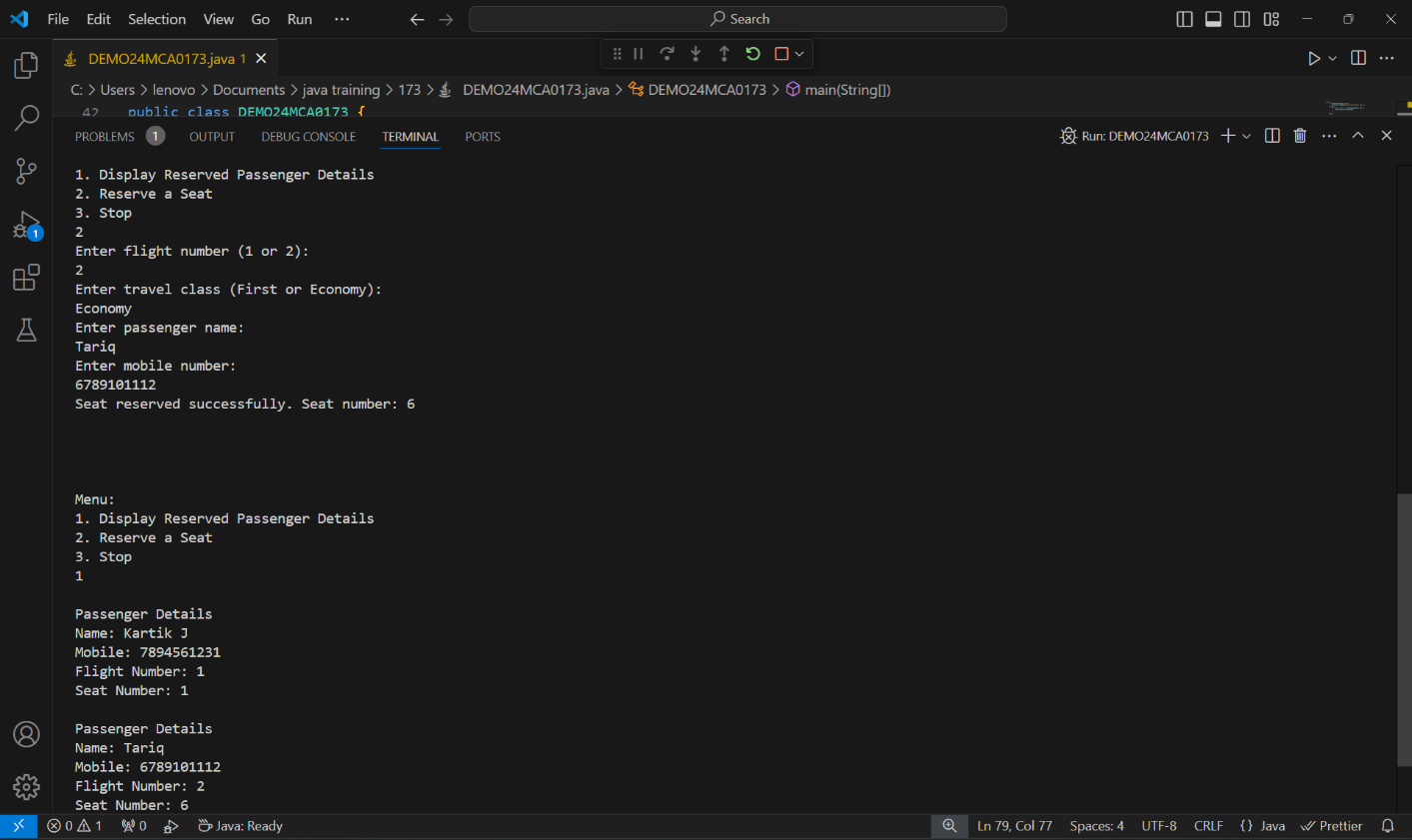
            }

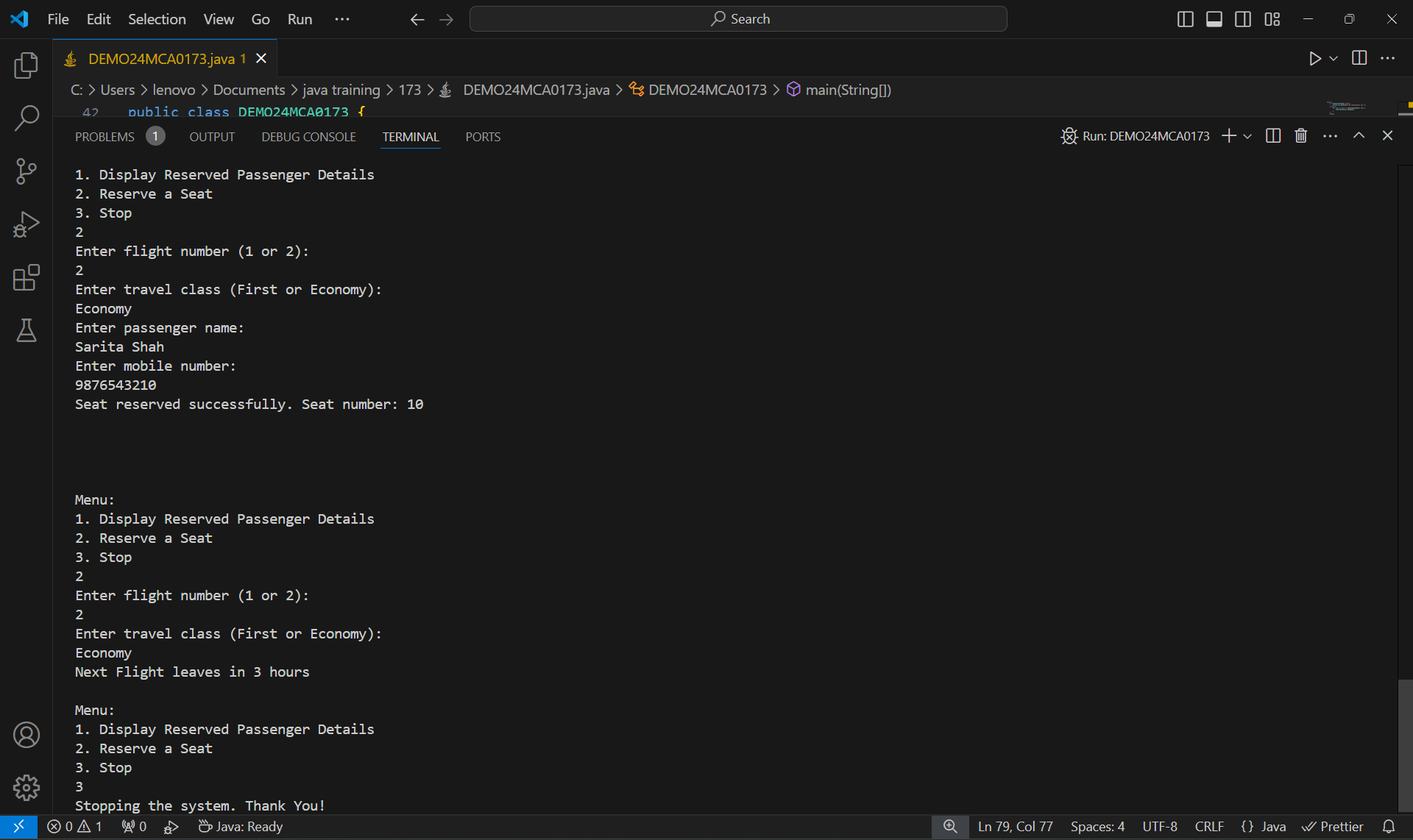
        }

    }

}







**PROGRAM-5**

**Understanding Strings**

**Some Websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rule is as follows:**

**A password must have at least eight characters.**

**A password consists of only letters and digits.**

**A password must contain at least two digits.**

**Write a program that prompts the user to enter a password and displays "Valid Password" if the rule is followed or "Invalid Password" otherwise.**

import java.util.Scanner;

public class UnderstandingStrings {

    public static boolean isvalid(String p){

        int len = p.length();

        if(len < 8) return false;

        int countDigit = 0;

        for(int i = 0; i < len; i++ ){

            char ch = p.charAt(i);

            if((ch  >= 'a' && ch <= 'z') || (ch  >= 'A' && ch <= 'Z')) continue;

            if(ch  >= '0' && ch <= '9') {countDigit++; continue;}

            return false;

        }

        if(countDigit < 2) return false;

        return true;

    }

    public static void main(String args[]){

        System.out.println("Enter Password");

        Scanner sc = new Scanner(System.in);

        String password = sc.next();

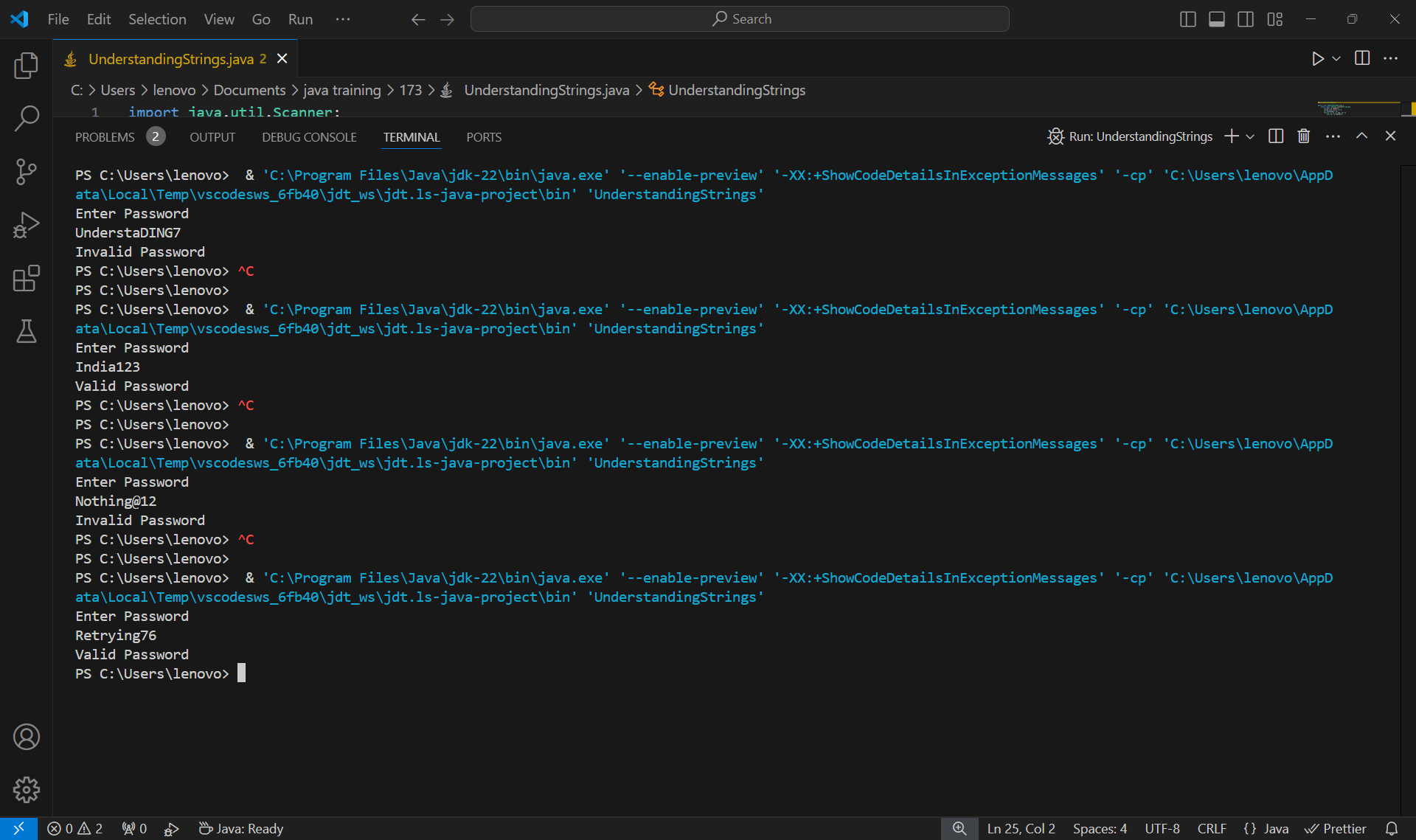
        if(isvalid(password) == true){

            System.out.println("Valid Password");

        }else System.out.print("Invalid Password");

    }

}



**PROGRAM-6**

**Understanding Inheritance**

**A company pays its employees on a weekly basis. The company has four types of employees: salaried employees, who are paid a fixed weekly salary regardless of the number of hours worked; hourly employees, who are paid by the hour and receive overtime pay; commission employees, who are paid a percentage of their sales; and salaried commission employees, who receive a base salary plus a percentage of their sales. For a current pay period, the company has decided to reward salaried commission employees by adding 10% to their salaries. The company wants to implement a java application that performs its payroll calculations polymorphically.**

abstract class Employee{

    float sal;

    public abstract float calSalary();

}

class Salaried extends Employee{

    Salaried(){sal = 0;}

    Salaried(float s){sal = s;}

    public float calSalary(){

        return sal;

    }

}

class Hourly extends Employee{

    int weekHour;

    Hourly(){sal = weekHour = 0;}

    Hourly(int h)

    {

        weekHour = h;

    }

    public float calSalary(){

        sal = weekHour\*100;

        //Overtime

        if(weekHour > 40){

            sal += ((weekHour-40)\*20);

        }

        return sal;

    }

}

class Commission extends Employee{

    float sales;

    Commission(){sal = sales = 0;}

    Commission(float sales){

        this.sales = sales;

    }

    public float calSalary(){

        sal = sales\*0.15f; //Paying 15% of Sales

        return sales;

    }

}

class SalariedCommision extends Employee{

    float base, sales;

    SalariedCommision(){

        base = sales = sal = 0;

    }

    SalariedCommision(float base, float sales){

        this.base = base;

        this.sales = sales;

    }

    public float calSalary(){

        sal = base + (sales\*0.15f); //receive a base salary plus a percentage of their sales

        sal += sal\*0.10f; //reward by adding 10% to their salaries

        return sal;

    }

}

public class UnderstandingInheritance {

    public static void main(String[] args) {

        Employee employeeRef;

        System.out.println("Payroll for Salaried Employee");

        employeeRef= new Salaried(500000);

        System.out.println("Salary is: "+employeeRef.calSalary());

        System.out.println("Payroll for Hourly Employee");

        employeeRef = new Hourly(50);

        System.out.println("Salary is: "+employeeRef.calSalary());

        System.out.println("Payroll for Commissioned Employee");

        employeeRef = new Commission(100000);

        System.out.println("Salary is: "+employeeRef.calSalary());

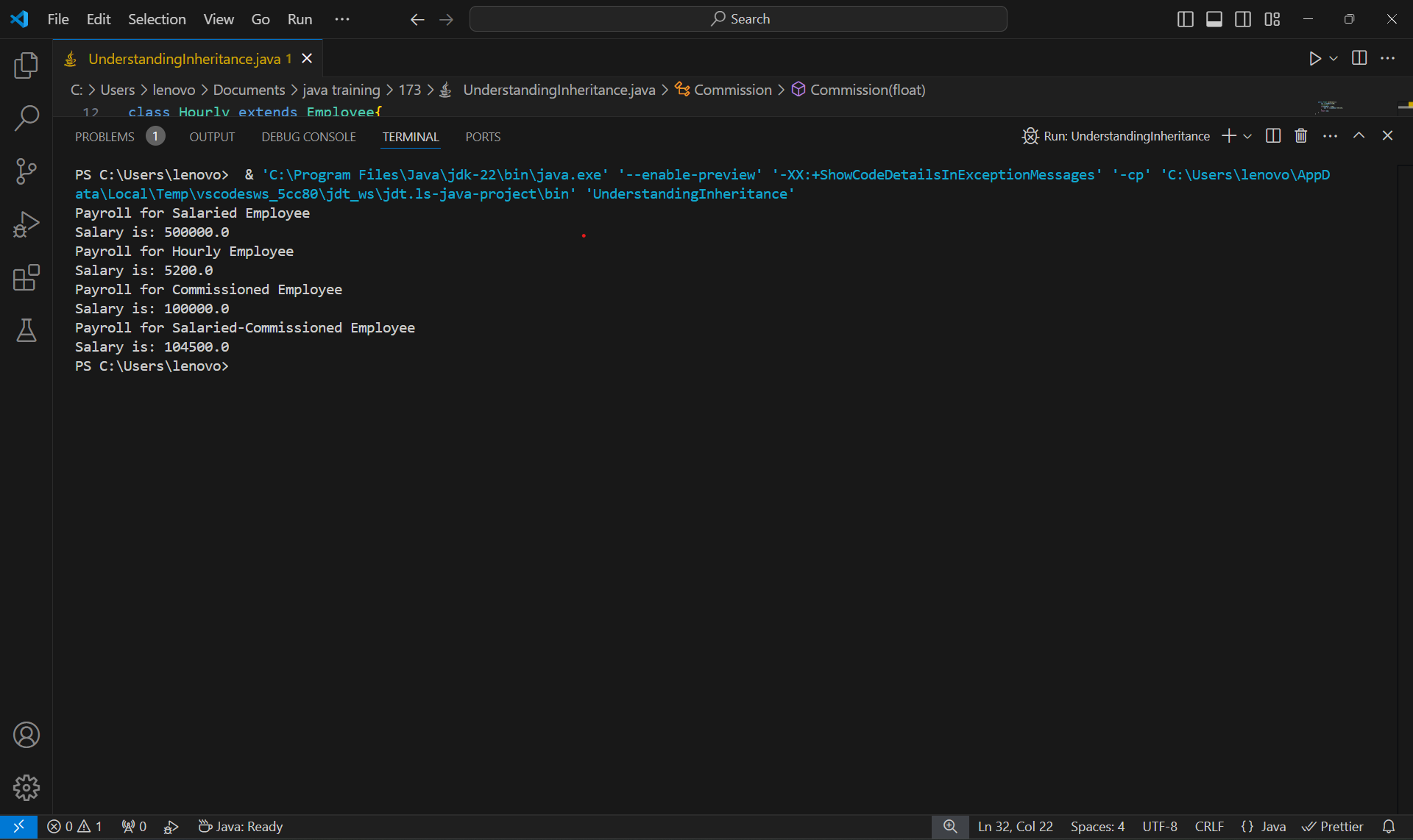
        System.out.println("Payroll for Salaried-Commissioned Employee");

        employeeRef = new SalariedCommision(20000, 500000);

        System.out.println("Salary is: "+employeeRef.calSalary());

    }

}



**PROGRAM-7**

**Understanding Interfaces**

**Develop a java program consisting of Shape hierarchy and a specialized Cylinder and Cone classes with appropriate functionality for computing the area of respective shape. Use Interfaces and implement runtime polymorphic behaviour in it.**

import java.util.\*;

interface Shape {

    float PI = 3.14f;

    public float area();

}

class Cylinder implements Shape{

    float radius;

    float height;

    Cylinder(float r, float h){

        radius = r;

        height = h;

    }

    public float area(){

        return (2\*PI\*radius\*(radius + height));

    }

}

class Cone implements Shape{

    float radius;

    float height;

    Cone(float r, float h){

        radius = r;

        height = h;

    }

    public float area(){

        return ((1.0f/3)\*PI\*radius\*(radius + height));

    }

}

public class UnderstandingInterfaces{

    public static void main(String[] args) {

        Cylinder cylinder = new Cylinder(1.4f, 7.2f);

        Cone cone = new Cone(5.5f, 8.3f);

        System.out.println("Area of Cylinder = "+ cylinder.area());

        System.out.println("Area of Cone = "+ cone.area());

    }

}

