**VIT,**

**VELLORE**

**PERIODIC ASSESMENT TEST – 1**

**JAVA PROGRAMMING LAB**

**(PMCA501P)**

**SUBMITTED TO:-**

**MR. SETHIL MURUGAN B**

**SCORE**

**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 Program1{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

Labour[] l = new Labour[3];

for(int i = 0; i < 3; i++){

System.out.println("Enter Labour details -");

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

String name = sc.nextLine();

System.out.print("Enter working hours: ");

int workingHours = sc.nextInt();

sc.nextLine(); // Consume the newline

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

l[i] = new Labour(name, workingHours);

l[i].calculateWages();

}

for(int i = 0; i < 3; i++){

l[i].display();

}

sc.close();

}

}

class Labour {

String name;

int workingHours;

float wage;

Labour(String name, int workingHours){

this.name = name;

this.workingHours = workingHours;

}

void calculateWages(){

if(workingHours > 40){

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

} else {

wage = workingHours \* 8.25f;

}

}

void display(){

System.out.println("Labour Details -");

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

System.out.println("Working Hours: " + workingHours);

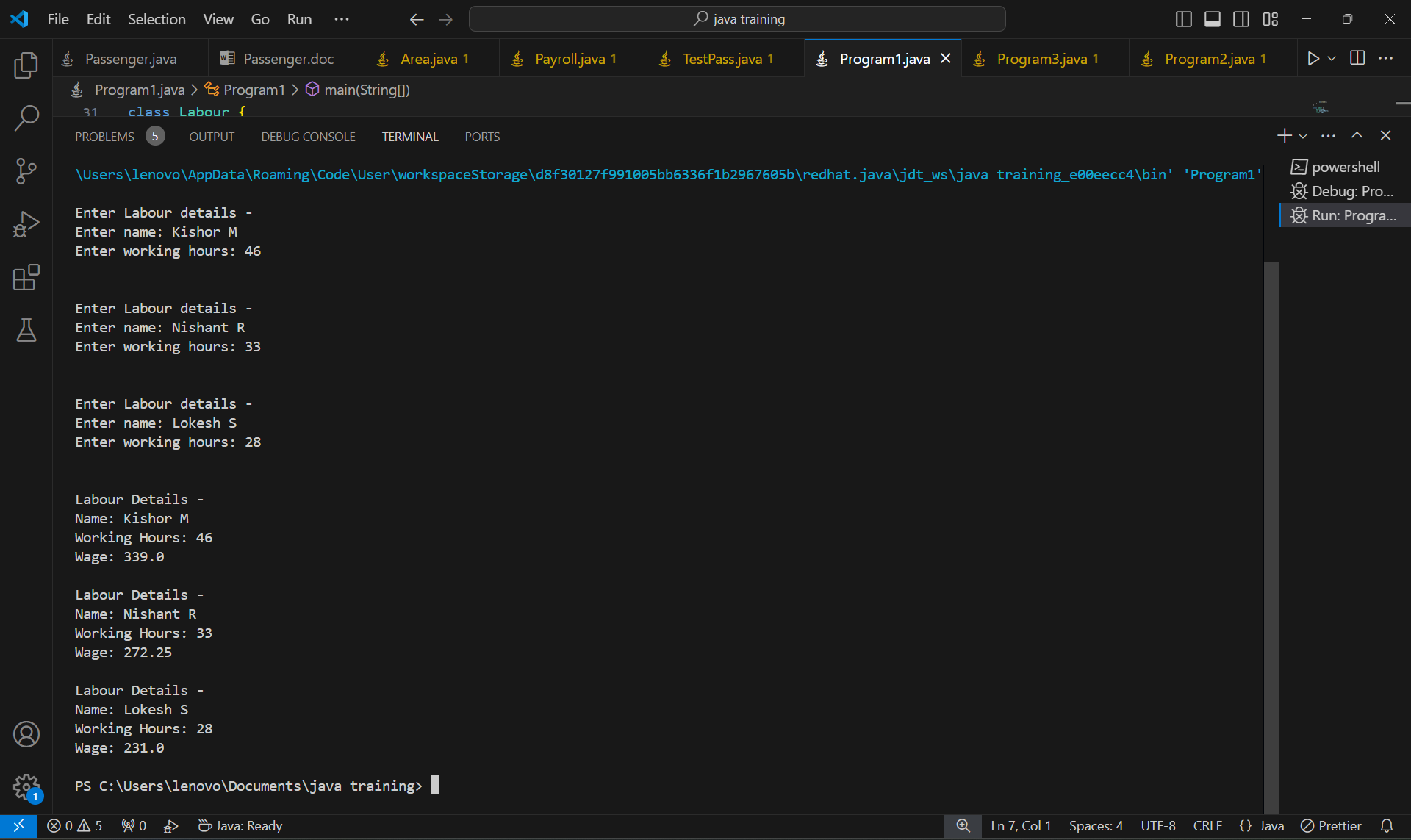
System.out.println("Wage: " + wage);

System.out.println("");

}

}

**OUTPUT:**



**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 Program2 {

public static void main(String[] args) {

isPrime();

}

public static void isPrime(){

int x, y=0;

Scanner sc = new Scanner(System.in);

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

x = sc.nextInt();

while(x > 0){

y = y\*10 + x%10;

x /= 10;

}

if (y <= 1) {

System.out.println(y + " is not prime");

return;

}

for (int i = 2; i <= y/2; i++) {

if (y % i == 0) {

System.out.println(y + " is not prime");

return;

}

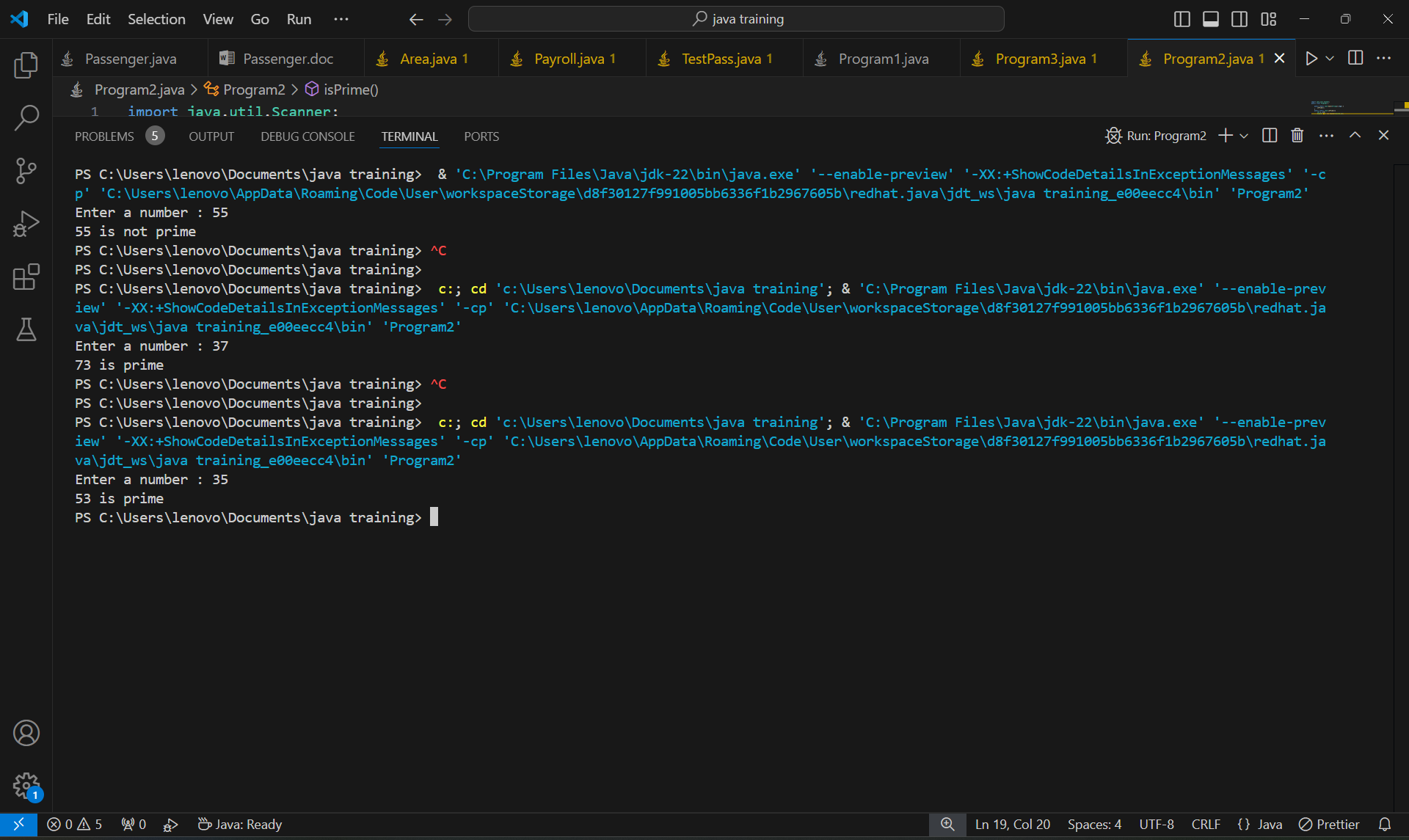
}

System.out.println(y +" is prime");

}

}

**OUTPUT:**



**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 Program3 {

public static void main(String args[]){

People[] p = new People[5];

for(int i=0; i<5; i++){

p[i] = new People();

p[i].putValues();

}

int i=0;

for(People x : p){

if(x.age>40){

i++;

}

}

System.out.println("\n\nThe total number of people above 40 are " + i);

}

}

class People{

public String name;

public int age;

public char gender;

public String qualification;

public float salary;

public void putValues(){

Scanner sc = new Scanner(System.in);

System.out.print("\nEnter name : ");

name = sc.next();

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

age = sc.nextInt();

System.out.print("Enter gender (M/F/O) : ");

gender = sc.next().charAt(0);

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

qualification = sc.nextLine();

sc.nextLine();

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

salary = sc.nextFloat();

}

public void getValues(){

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

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

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

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

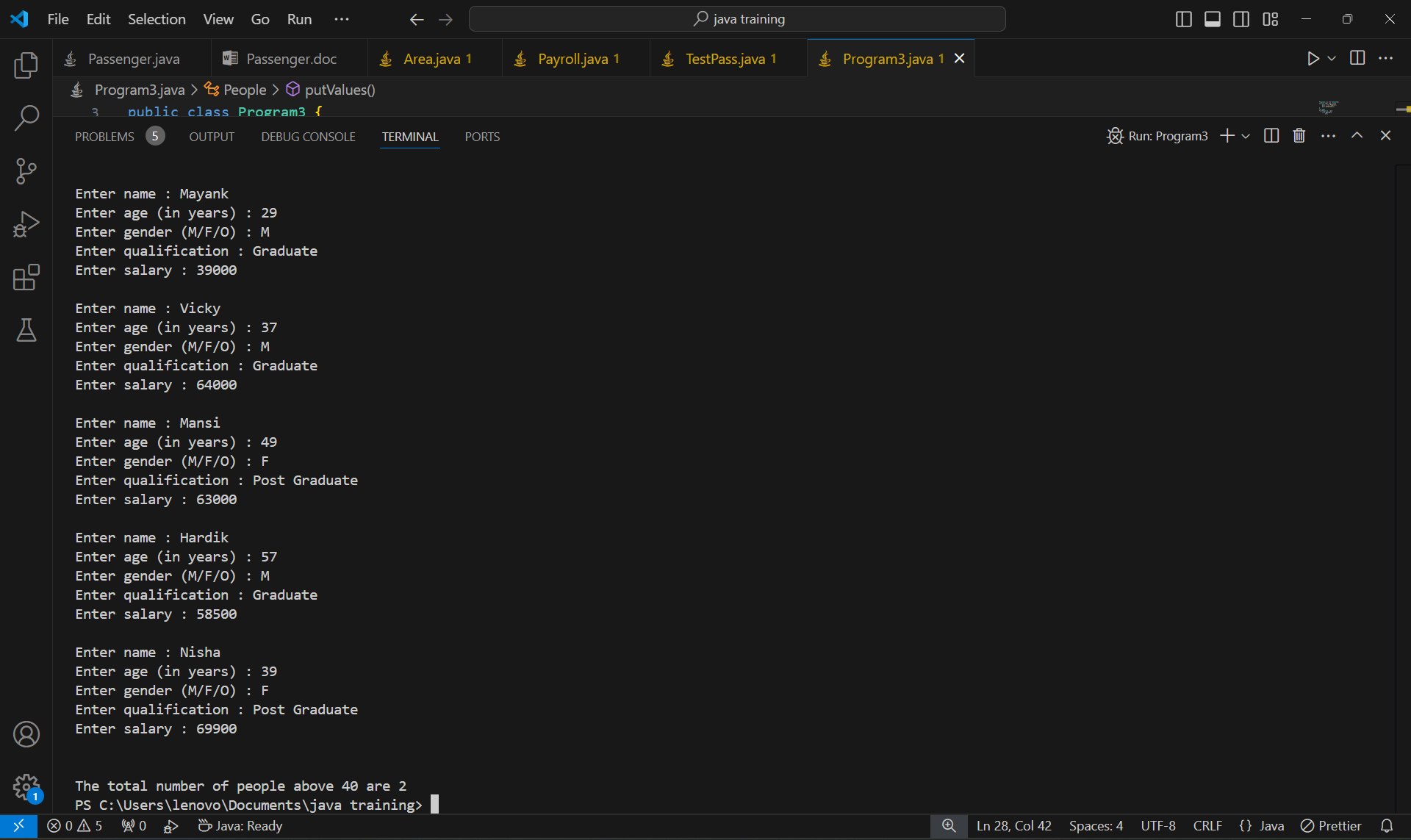
System.out.print("Salary : " + salary);

System.out.println("");

}

}

**OUTPUT:**



**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.**

// AIRLINES

import java.util.Scanner;

public class Passenger {

String name;

String mobilenumber;

int flightNo;

int seatNo;

String type;

static String[] flight1 = {"Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available"};

static String[] flight2 = {"Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available", "Available"};

public Passenger(String name, String mno) {

this.name = name;

this.mobilenumber = mno;

this.flightNo = -1;

this.seatNo = -1;

this.type = "N/A";

}

public void book(int flno, char type, Scanner sc) { // Pass Scanner object from main

if (flno == 1) {

if (getSeat(flight1, type) >= 0) {

char ch;

System.out.print("\nSeat is Available...\nDo you want to proceed[Y/N]... ");

ch = sc.next().charAt(0);

sc.nextLine();

if (ch == 'Y' || ch == 'y') {

if(type=='F'){

this.type = "First";

}

else if(type == 'E'){

this.type = "Economy";

}

seatNo = getSeat(flight1, type);

flightNo = flno;

flight1[seatNo] = "Reserved";

System.out.println("Booking Confirmed !!");

System.out.println("Flight No : " + flno + "\t" + "Seat No : " + seatNo);

} else {

System.out.println("Thank you... Please visit again...");

}

} else {

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

System.out.println("Thank you... Please visit again...");

}

} else if (flno == 2) {

if (getSeat(flight2, type) >= 0) {

char ch;

System.out.print("\nSeat is Available...\nDo you want to proceed[Y/N]... ");

ch = sc.next().charAt(0);

sc.nextLine();

if (ch == 'Y' || ch == 'y') {

if(type=='F'){

this.type = "First";

}

else if(type == 'E'){

this.type = "Economy";

}

seatNo = getSeat(flight2, type);

flightNo = flno;

flight2[seatNo] = "Reserved";

System.out.println("Booking Confirmed !!");

System.out.println("Flight No : " + flno + "\t" + "Seat No : " + seatNo);

} else {

System.out.println("Thank you... Please visit again...");

}

} else {

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

System.out.println("Thank you... Please visit again...");

}

}

}

public void getDetails() {

System.out.println("\n\*\*\*\*\*Passenger Details\*\*\*\*\*\nName : " + name + "\nMobile No : " + mobilenumber);

System.out.println("Flight No : " + flightNo + "\nSeat Type : " + type + "\nSeat No : " + seatNo + "\n\t\*\*\*\*\*\*\*\*\*\*\*\*");

}

int getSeat(String[] fl, char type) {

if (type == 'E') {

for (int i = 5; i <= 9; i++) {

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

return i;

}

}

} else if (type == 'F') {

for (int i = 0; i <= 4; i++) {

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

return i;

}

}

}

return -1;

}

}

class Demo {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

Passenger[] pas = new Passenger[20];

int ch;

String name, mno;

for (int i = 0; i < 20; i++) {

System.out.println("\n\n\tPASSENGER REGISTRATION MENU");

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

name = sc.nextLine();

System.out.print("Enter your mobile no : ");

mno = sc.nextLine();

pas[i] = new Passenger(name, mno);

do {

System.out.println("\n\tM\tE\tN\tU");

System.out.println("1. Display Reserved Passenger Details\n2. Reserve a seat\n3. Stop");

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

ch = sc.nextInt();

sc.nextLine();

switch (ch) {

case 1:

pas[i].getDetails();

break;

case 2:

int fno;

char ty;

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

fno = sc.nextInt();

sc.nextLine();

System.out.print("Enter seat type (E for Economy or F for First) : ");

ty = sc.nextLine().charAt(0);

pas[i].book(fno, ty, sc);

break;

case 3:

System.out.println("Stopping registration.");

break;

default:

System.out.println("Please try again !!");

break;

}

} while (ch != 3);

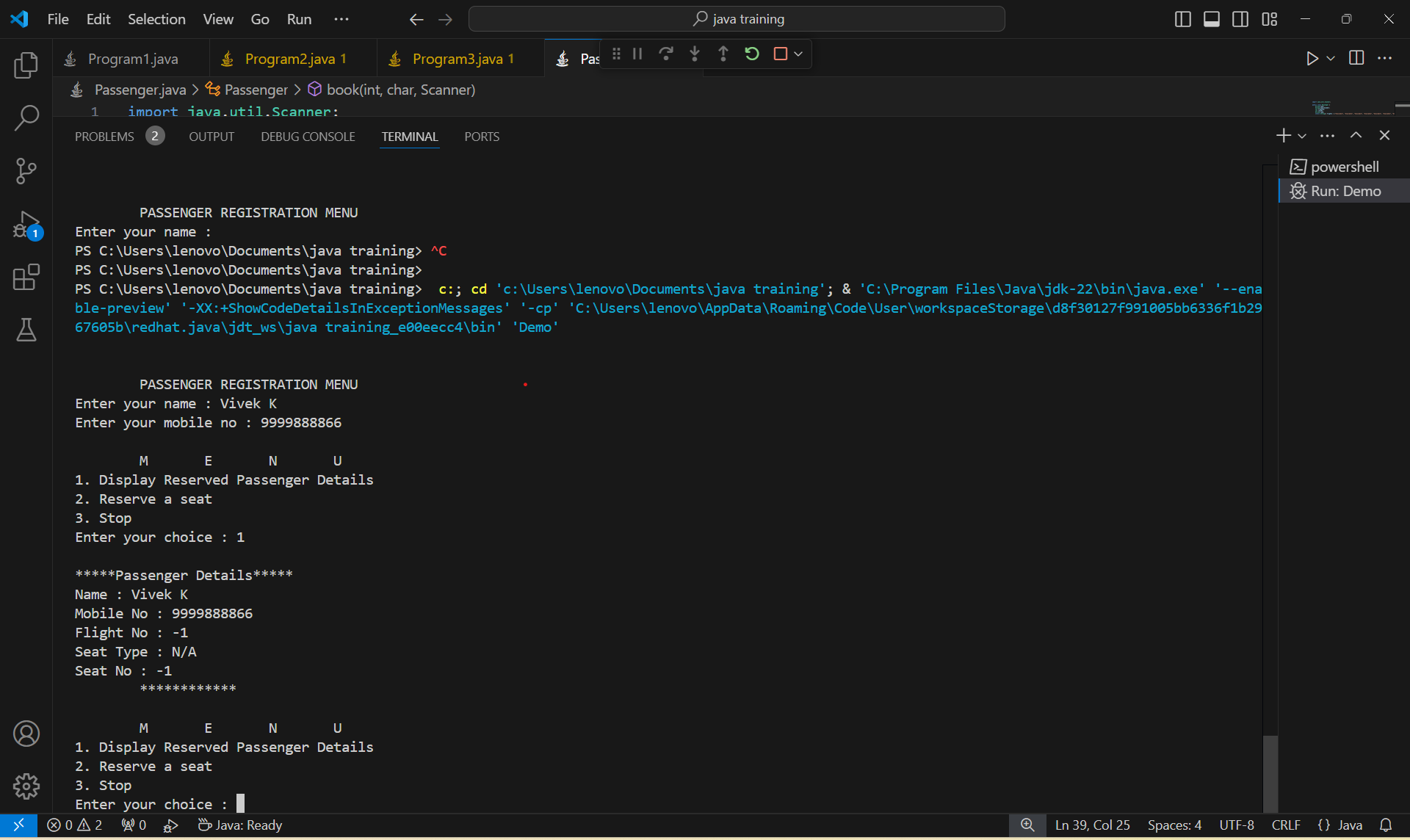
}

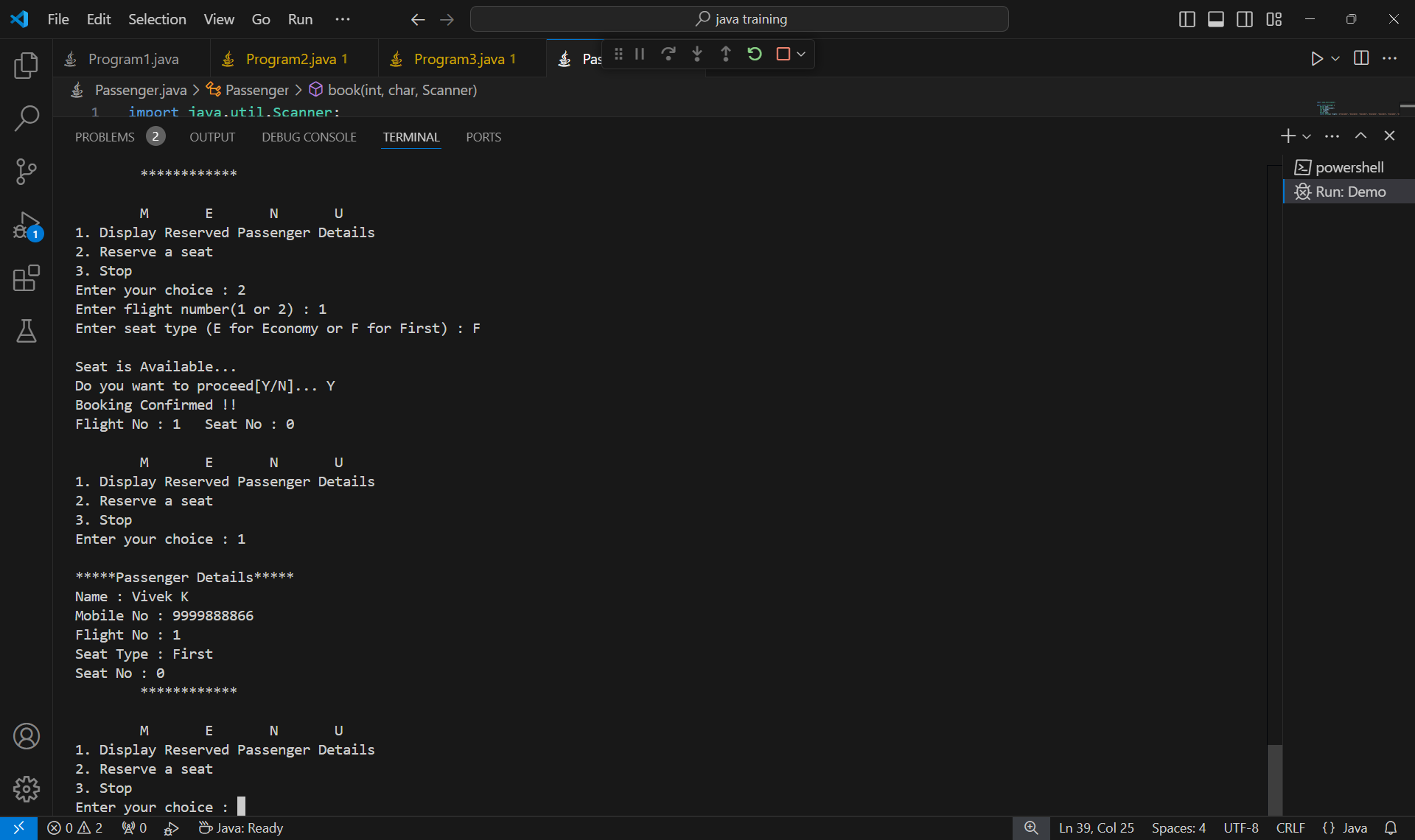
sc.close();

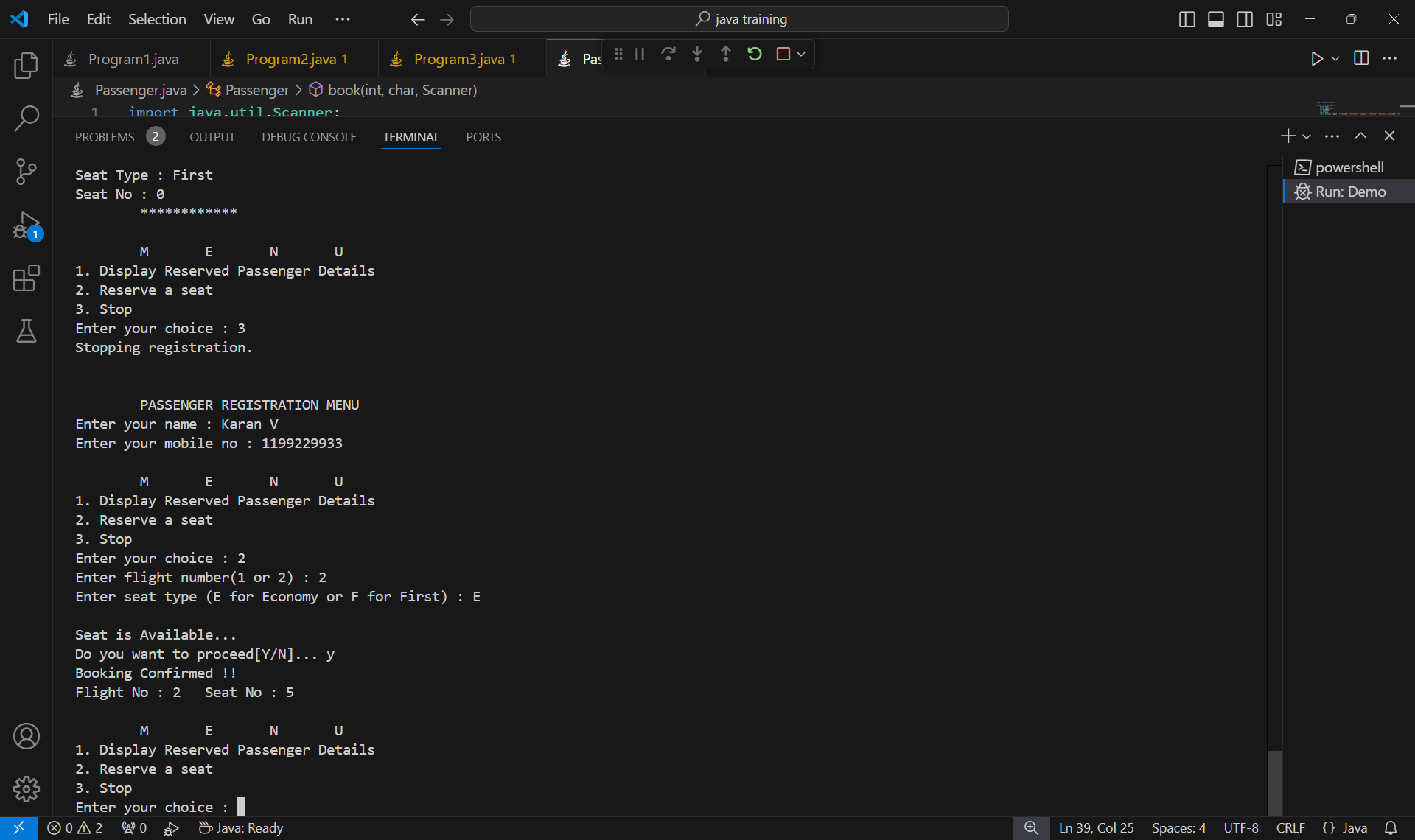
}

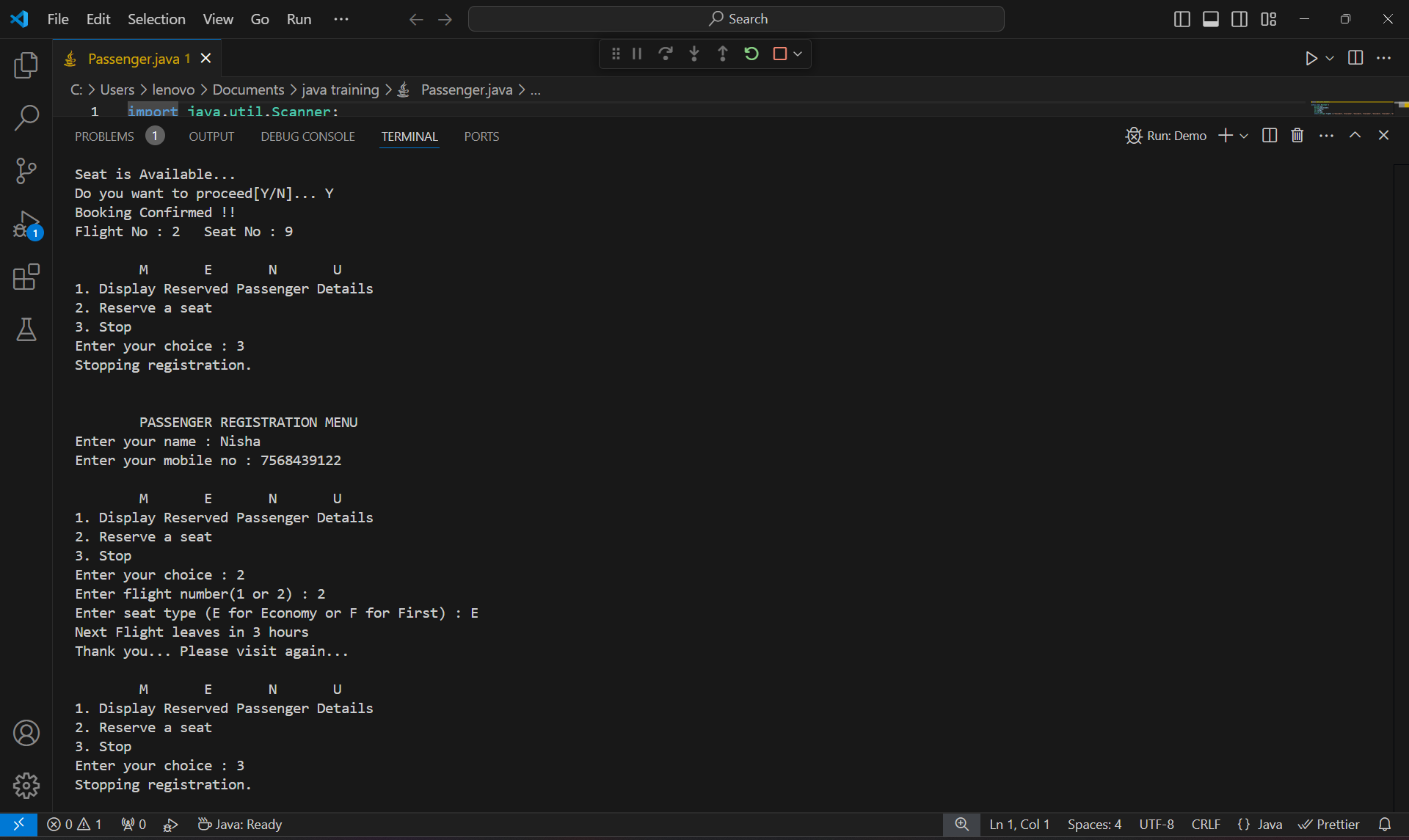
}

**OUTPUTS:**









**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 TestPass {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String pass;

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

        pass = sc.next();

        if(checkPass(pass)==true){

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

        }

        else{

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

        }

    }

    static boolean checkPass(String str){

        int num=0;

        if(str.length()>=8){

            for(char c : str.toCharArray()){

                if((c>='a' && c<'z') || (c>='A' && c<'Z')){

                    continue;

                }

                else if(c>='0' && c<='9'){

                    num++;

                    continue;

                }

                return false;

            }

        }

        if(num>=2){

            return true;

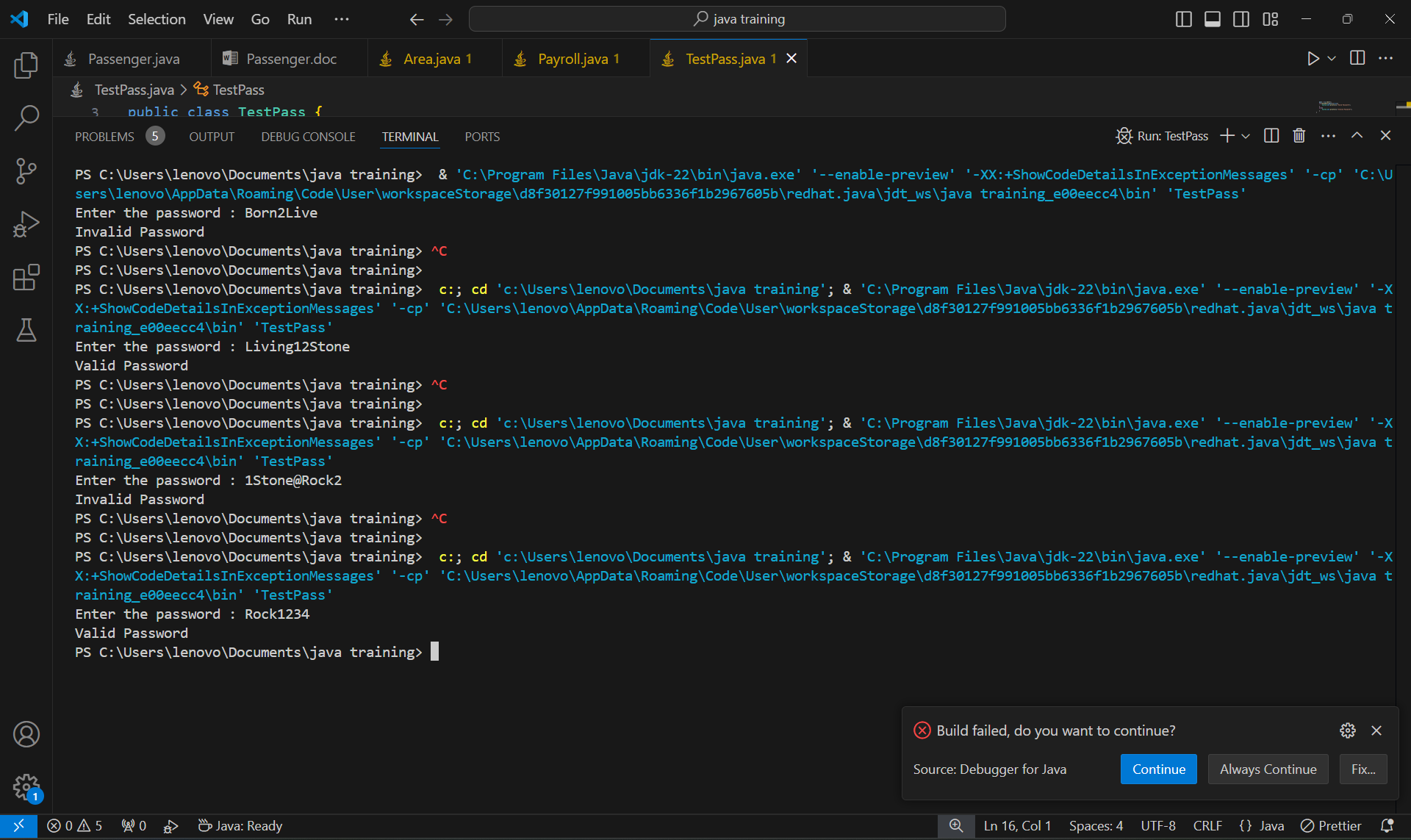
        }

        return false;

    }

}

**OUTPUT:**



**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.**

import java.util.\*;

public class Payroll {

    public static void main(String[] args) {

        Employee emp;

        emp = new SalariedEmp(31000.0f);

        System.out.println("Salaried Employee\nSalary = "+ emp.calculateSalary()+'\n');

        emp = new HourlyEmp(70);

        System.out.println("Hourly Employee\nSalary = "+ emp.calculateSalary()+'\n');

        emp = new ComissionEmp(115000.5f);

        System.out.println("Commission Employee\nSalary = "+ emp.calculateSalary()+'\n');

        emp = new SalariedCommEmp(136000.8f, 10000.0f);

        System.out.println("Salaried Commission Employee\nSalary = "+ emp.calculateSalary()+'\n');

    }

}

abstract class Employee {

    public float pay;

    public abstract float calculateSalary();

}

class SalariedEmp extends Employee{

    SalariedEmp(){

        pay = 0.0f;

    }

    SalariedEmp(float p){

        pay = p;

    }

    public float calculateSalary(){

        return pay;

    }

}

class HourlyEmp extends Employee{

    int weeklyHour;

    HourlyEmp(){

        pay = 0.0f;

        weeklyHour = 0;

    }

    HourlyEmp(int hr){

        weeklyHour = hr;

    }

    public float calculateSalary(){

        pay = weeklyHour \* 250;

        if(weeklyHour>40){

            pay += ((weeklyHour-40)\*100);

        }

        return pay;

    }

}

class ComissionEmp extends Employee{

    float sales;

    ComissionEmp(){

        pay = 0.0f;

        sales = 0.0f;

    }

    ComissionEmp(float sl){

        sales = sl;

    }

    public float calculateSalary(){

        pay = sales \*  0.15f;

        return pay;

    }

}

class SalariedCommEmp extends Employee{

    float baseSal, sales;

    SalariedCommEmp(){

        pay = 0.0f;

        sales = 0.0f;

        baseSal = 0.0f;

    }

    SalariedCommEmp(float sales, float base){

        this.sales = sales;

        baseSal = base;

    }

    public float calculateSalary(){

        pay = baseSal + (sales \* 0.15f);

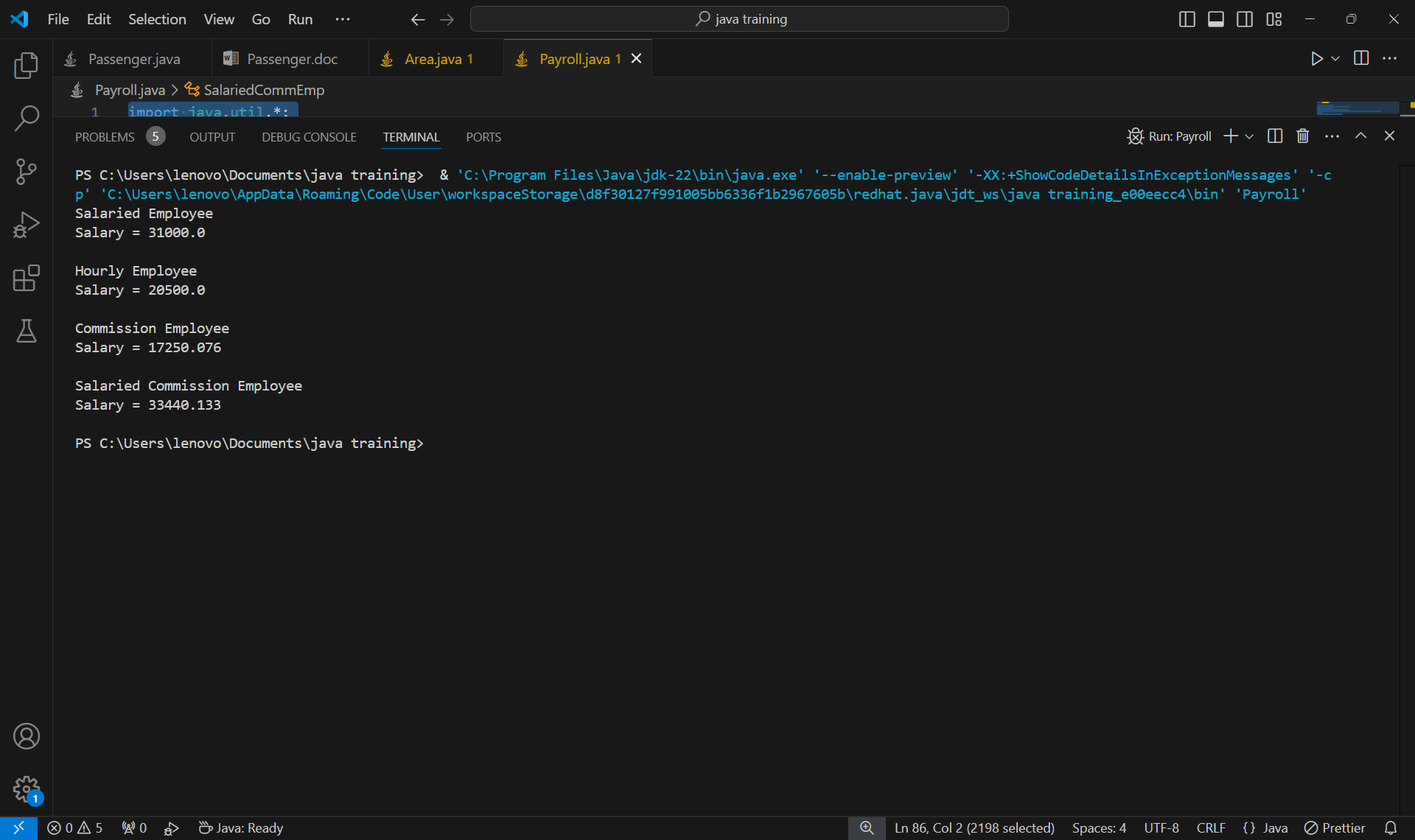
        pay += (pay \* 0.1f);

        return pay;

    }

}

**OUTPUT:**



**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 Area{

    public static void main(String[] args) {

        Cylinder cyl = new Cylinder(2.4f, 5.2f);

        Cone con = new Cone(3.5f, 5.3f);

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

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

    }

}

**OUTPUT:**

