Assignment No. 4 & 5

Class, Object, Constructor & Constructor chaining

1.Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student.

```java
/*1.Create a class named 'Student' with String variable 'name' and integer varia
the value of roll_no as '2' and that of name as "John" by creating an object of
class Student {
    String name;
    int roll_no;
    public void setValue(String a, int b) {
        name = a;
        roll_no = b;
    }
    public void display() {
        System.out.println("Name = " + name + " & Roll Number = " + roll_no);
    }
}
public class Que1 {

    Run main | Debug main
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.setValue("John", 2);
        s1.display();
    }
}
```

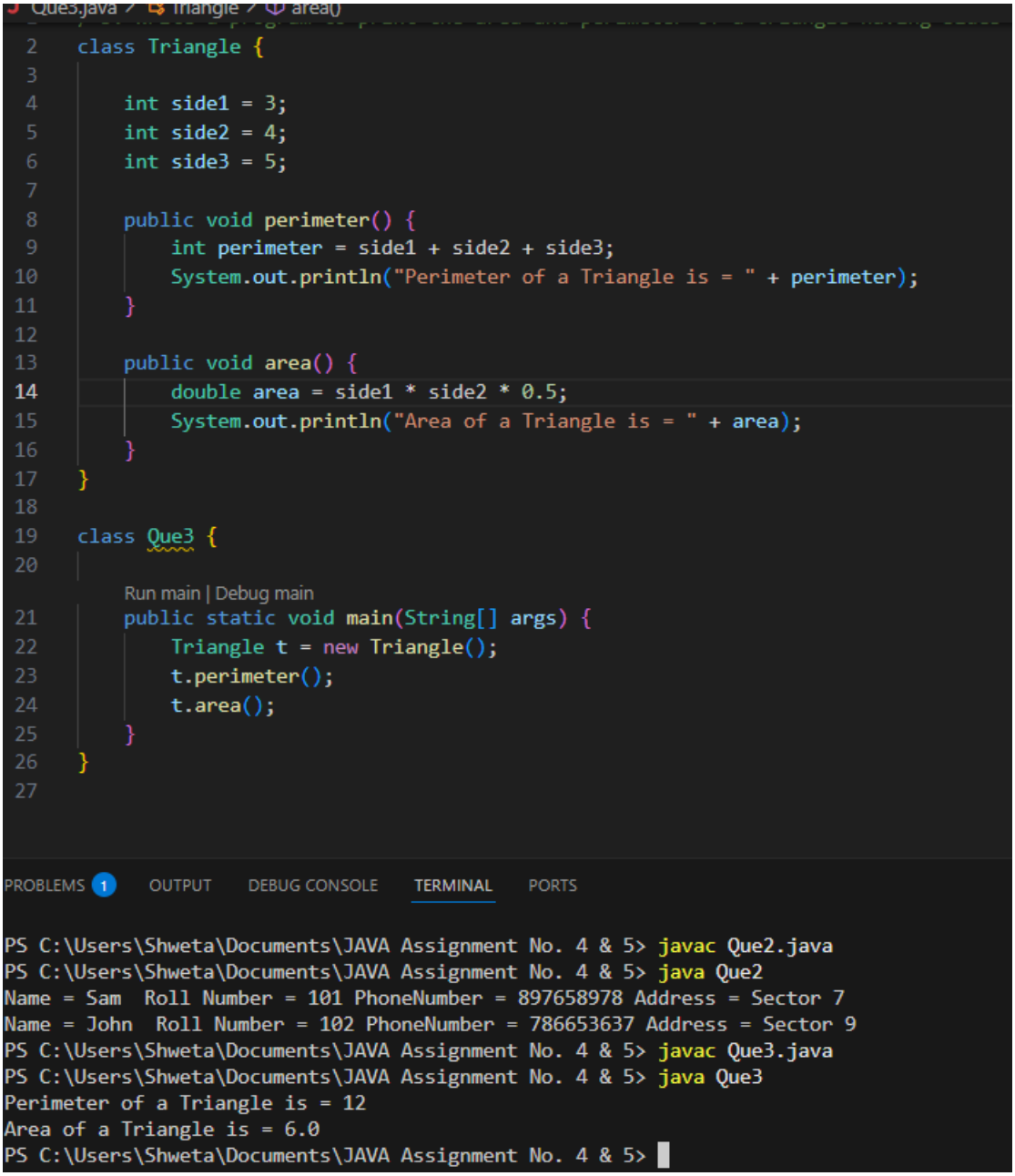
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que1.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que1
Name = John & Roll Number = 2
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

2. Assign and print the roll number, phone number and address of two students having names "Sam" and "John" respectively by creating two objects of class 'Student'.

```java
/*2. Assign and print the roll number, phone number and address of two students having names
"Sam" and "John" respectively by creating two objects of class 'Student'.*/
class Student {

    String name;
    int roll_no;
    long phone_no;
    String address;

    public void setValue(String n, int r, long p, String a) {
        name = n;
        roll_no = r;
        phone_no = p;
        address = a;
    }

    public void display() {
        System.out.println("Name = " + name + "  Roll Number = " + roll_no + " PhoneNumber = " + phone_no + " Address = " + address);
    }
}

public class Que2 {

    Run main | Debug main
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student();
        s1.setValue("Sam", 101, 89765897888L, "Sector 7");
        s2.setValue("John", 102, 7866536377L, "Sector 9");
        s1.display();
        s2.display();
    }
}
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Name = Sam  Roll Number = 101 PhoneNumber = 89765897888 Address = Sector 7
Name = John  Roll Number = 102 PhoneNumber = 7866536377 Address = Sector 9
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

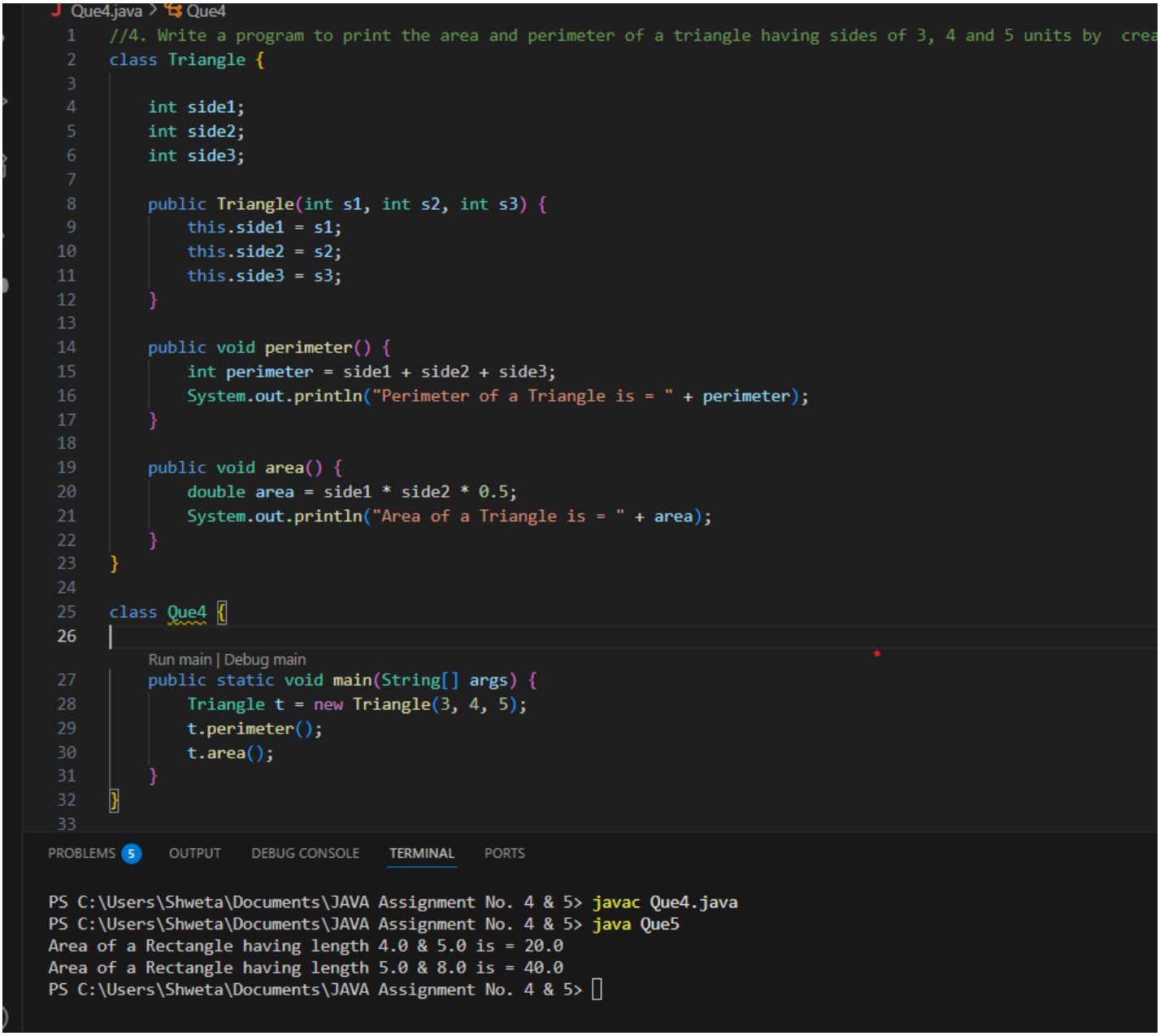
3. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.



```java
class Triangle {

    int side1 = 3;
    int side2 = 4;
    int side3 = 5;

    public void perimeter() {
        int perimeter = side1 + side2 + side3;
        System.out.println("Perimeter of a Triangle is = " + perimeter);
    }

    public void area() {
        double area = side1 * side2 * 0.5;
        System.out.println("Area of a Triangle is = " + area);
    }
}

class Que3 {

    Run main | Debug main
    public static void main(String[] args) {
        Triangle t = new Triangle();
        t.perimeter();
        t.area();
    }
}
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que2.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que2
Name = Sam  Roll Number = 101 PhoneNumber = 897658978 Address = Sector 7
Name = John  Roll Number = 102 PhoneNumber = 786653637 Address = Sector 9
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que3.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que3
Perimeter of a Triangle is = 12
Area of a Triangle is = 6.0
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

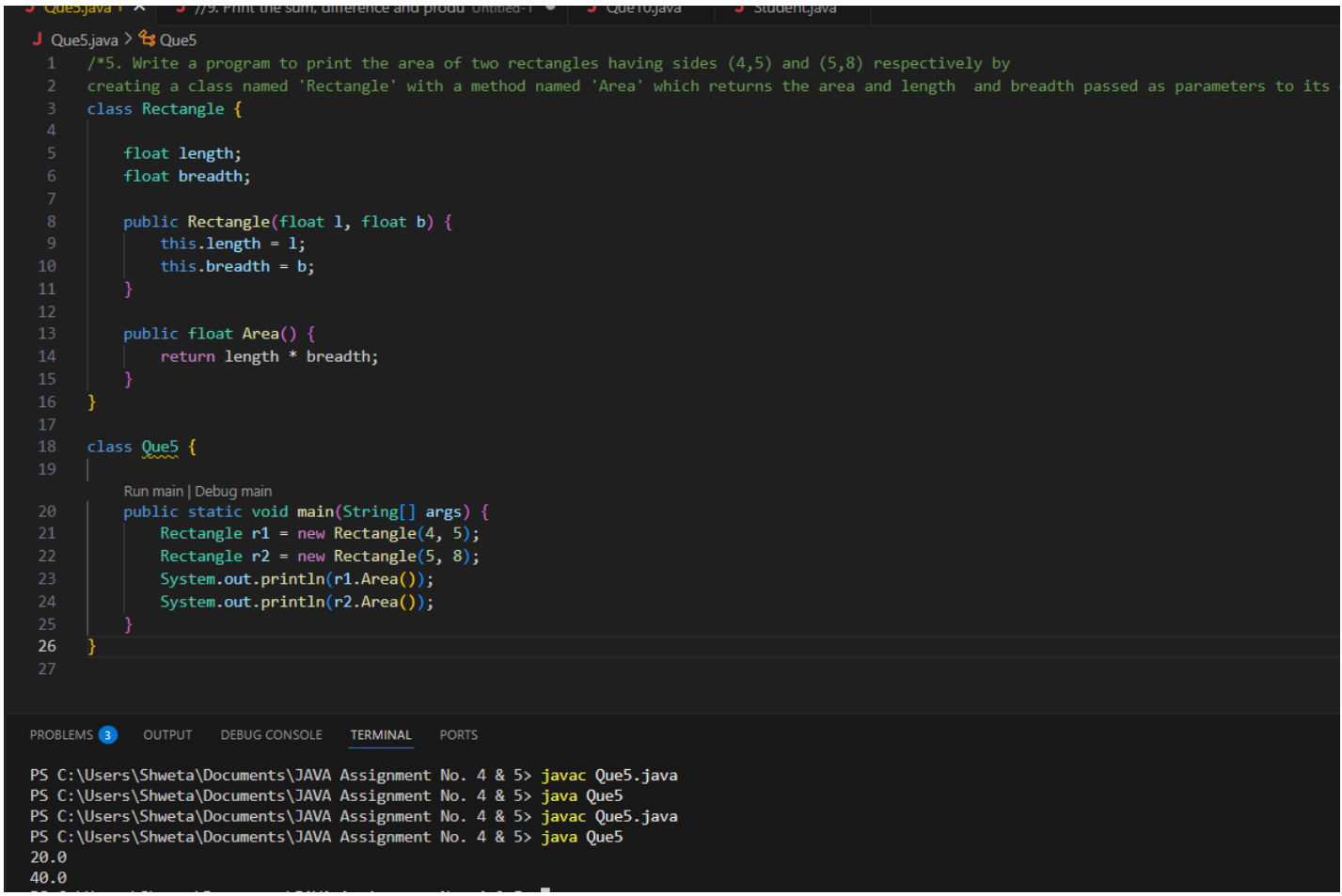
4. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' with the constructor having the three sides as its parameters.



```java
//4. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by  crea
class Triangle {

    int side1;
    int side2;
    int side3;

    public Triangle(int s1, int s2, int s3) {
        this.side1 = s1;
        this.side2 = s2;
        this.side3 = s3;
    }

    public void perimeter() {
        int perimeter = side1 + side2 + side3;
        System.out.println("Perimeter of a Triangle is = " + perimeter);
    }

    public void area() {
        double area = side1 * side2 * 0.5;
        System.out.println("Area of a Triangle is = " + area);
    }
}

class Que4 {

    Run main | Debug main
    public static void main(String[] args) {
        Triangle t = new Triangle(3, 4, 5);
        t.perimeter();
        t.area();
    }
}
```

```
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que4.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que5
Area of a Rectangle having length 4.0 & 5.0 is = 20.0
Area of a Rectangle having length 5.0 & 8.0 is = 40.0
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

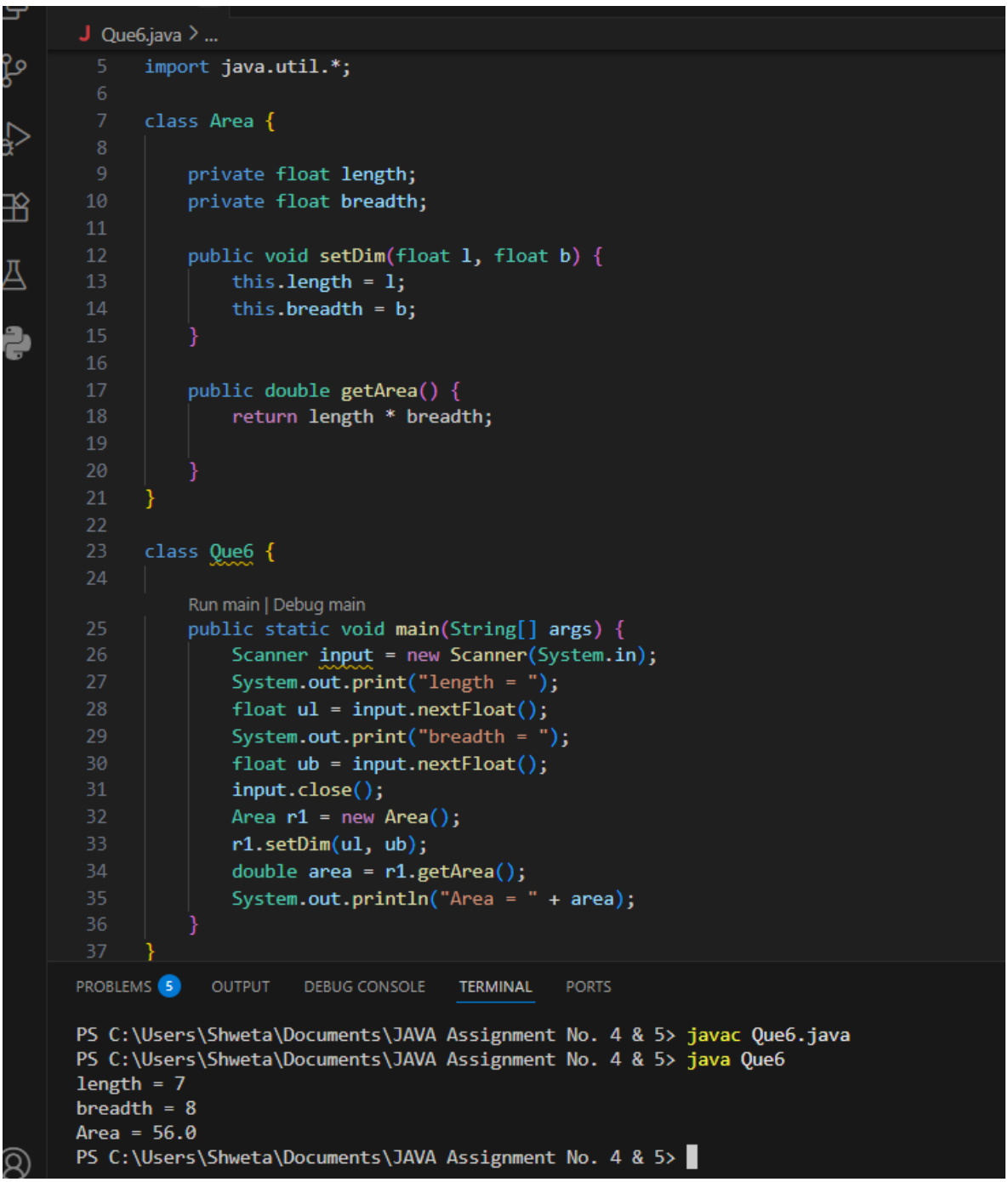
5. Write a program to print the area of two rectangles having sides (4,5) and (5,8) respectively by creating a class named 'Rectangle' with a method named 'Area' which returns the area and length and breadth passed as parameters to its constructor.



```java
/*5. Write a program to print the area of two rectangles having sides (4,5) and (5,8) respectively by
creating a class named 'Rectangle' with a method named 'Area' which returns the area and length  and breadth passed as parameters to its
class Rectangle {

    float length;
    float breadth;

    public Rectangle(float l, float b) {
        this.length = l;
        this.breadth = b;
    }

    public float Area() {
        return length * breadth;
    }
}

class Que5 {

    Run main | Debug main
    public static void main(String[] args) {
        Rectangle r1 = new Rectangle(4, 5);
        Rectangle r2 = new Rectangle(5, 8);
        System.out.println(r1.Area());
        System.out.println(r2.Area());
    }
}
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que5.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que5
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que5.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que5
20.0
40.0
```

6. Write a program to print the area of a rectangle by creating a class named 'Area' having two methods. First method named as 'setDim' takes length and breadth of the rectangle as parameters and the second method named as 'getArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through the keyboard.



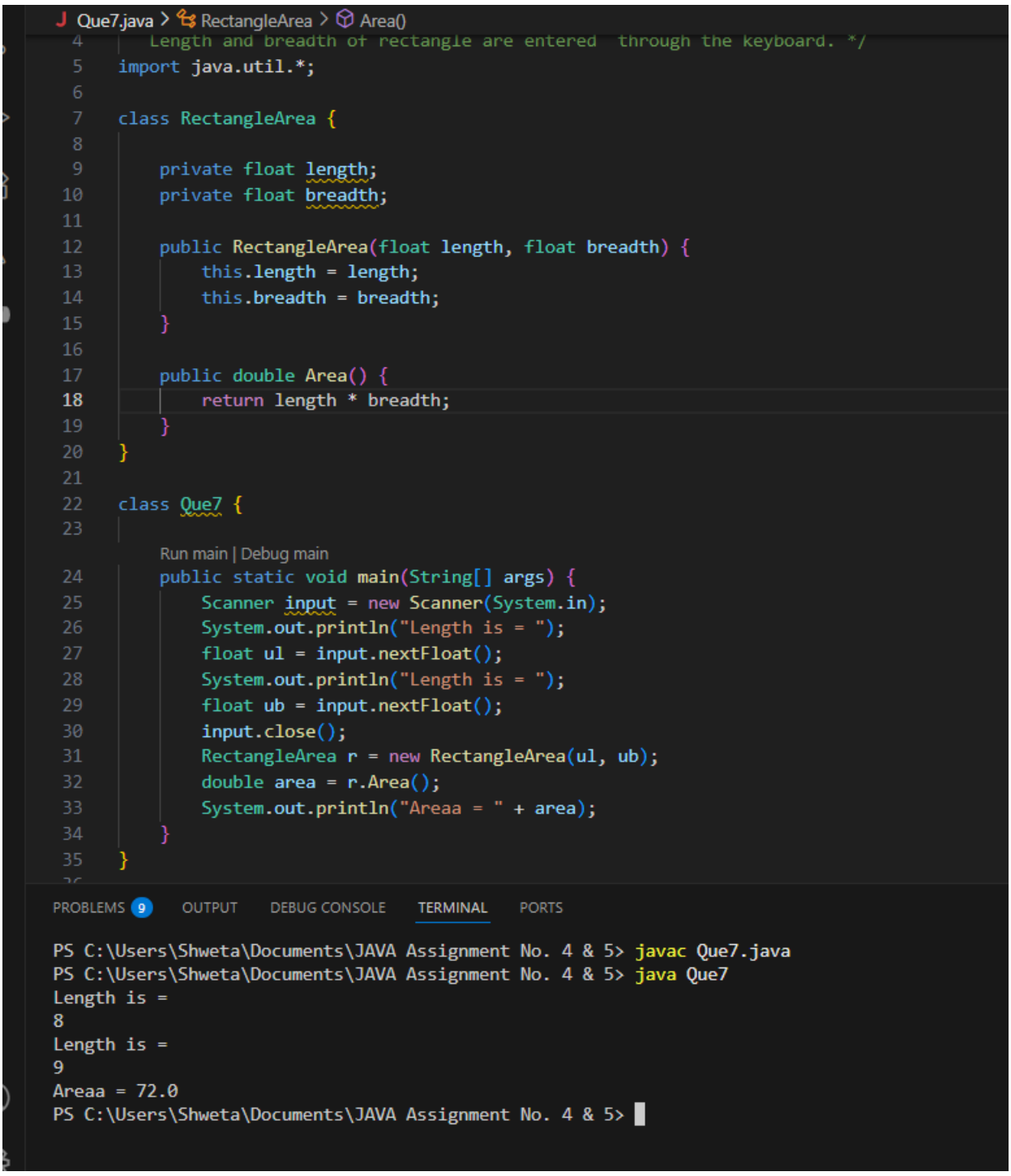
Que6.java > ...

```java
import java.util.*;

class Area {

    private float length;
    private float breadth;

    public void setDim(float l, float b) {
        this.length = l;
        this.breadth = b;
    }

    public double getArea() {
        return length * breadth;

    }
}

class Que6 {

    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("length = ");
        float ul = input.nextFloat();
        System.out.print("breadth = ");
        float ub = input.nextFloat();
        input.close();
        Area r1 = new Area();
        r1.setDim(ul, ub);
        double area = r1.getArea();
        System.out.println("Area = " + area);
    }
}
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que6.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que6
length = 7
breadth = 8
Area = 56.0
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

7. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through the keyboard.



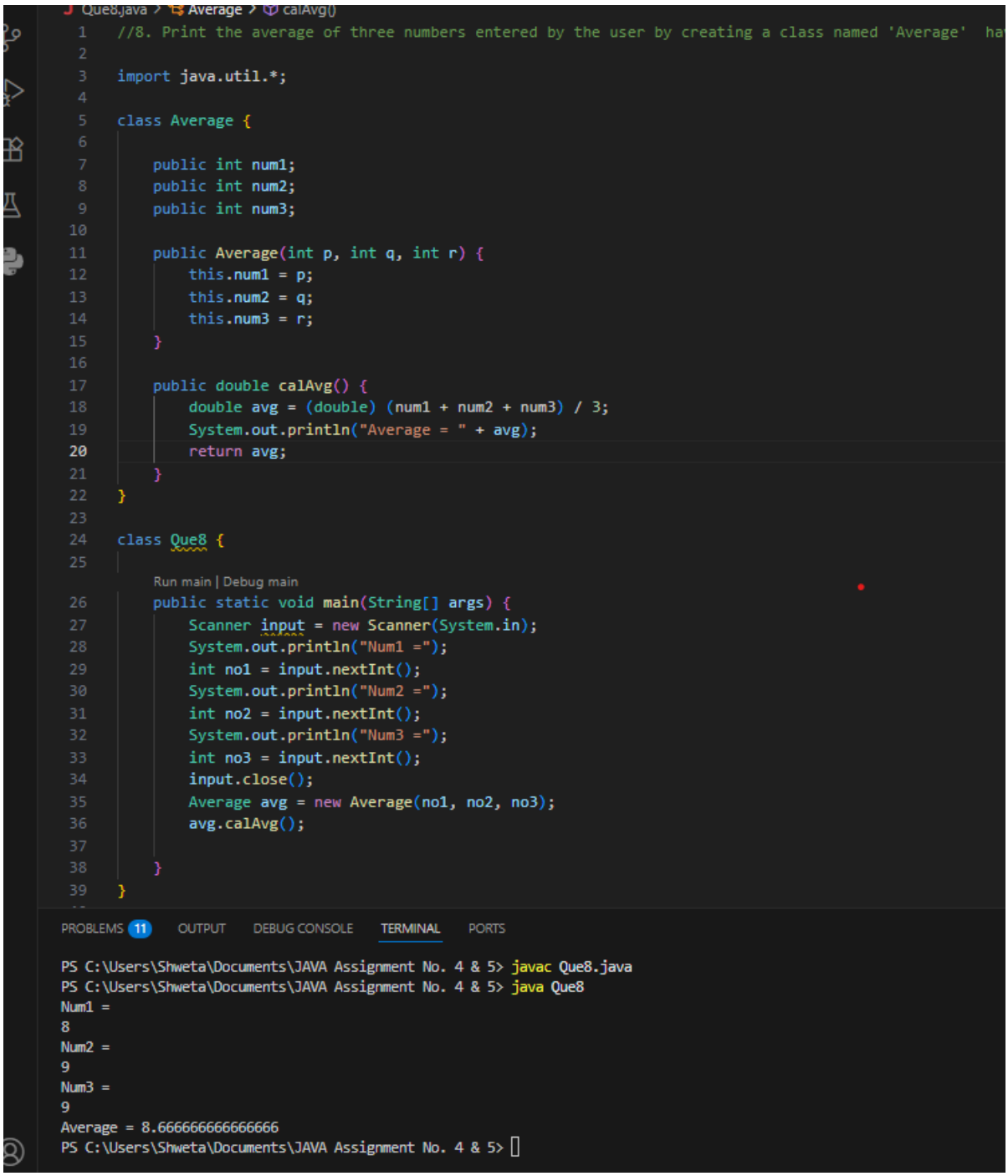
Que7.java > RectangleArea > Area()

```
   Length and breadth of rectangle are entered  through the keyboard. */
import java.util.*;

class RectangleArea {

    private float length;
    private float breadth;

    public RectangleArea(float length, float breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double Area() {
        return length * breadth;
    }
}

class Que7 {

    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Length is = ");
        float ul = input.nextFloat();
        System.out.println("Length is = ");
        float ub = input.nextFloat();
        input.close();
        RectangleArea r = new RectangleArea(ul, ub);
        double area = r.Area();
        System.out.println("Areaa = " + area);
    }
}
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que7.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que7
Length is =
8
Length is =
9
Areaa = 72.0
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

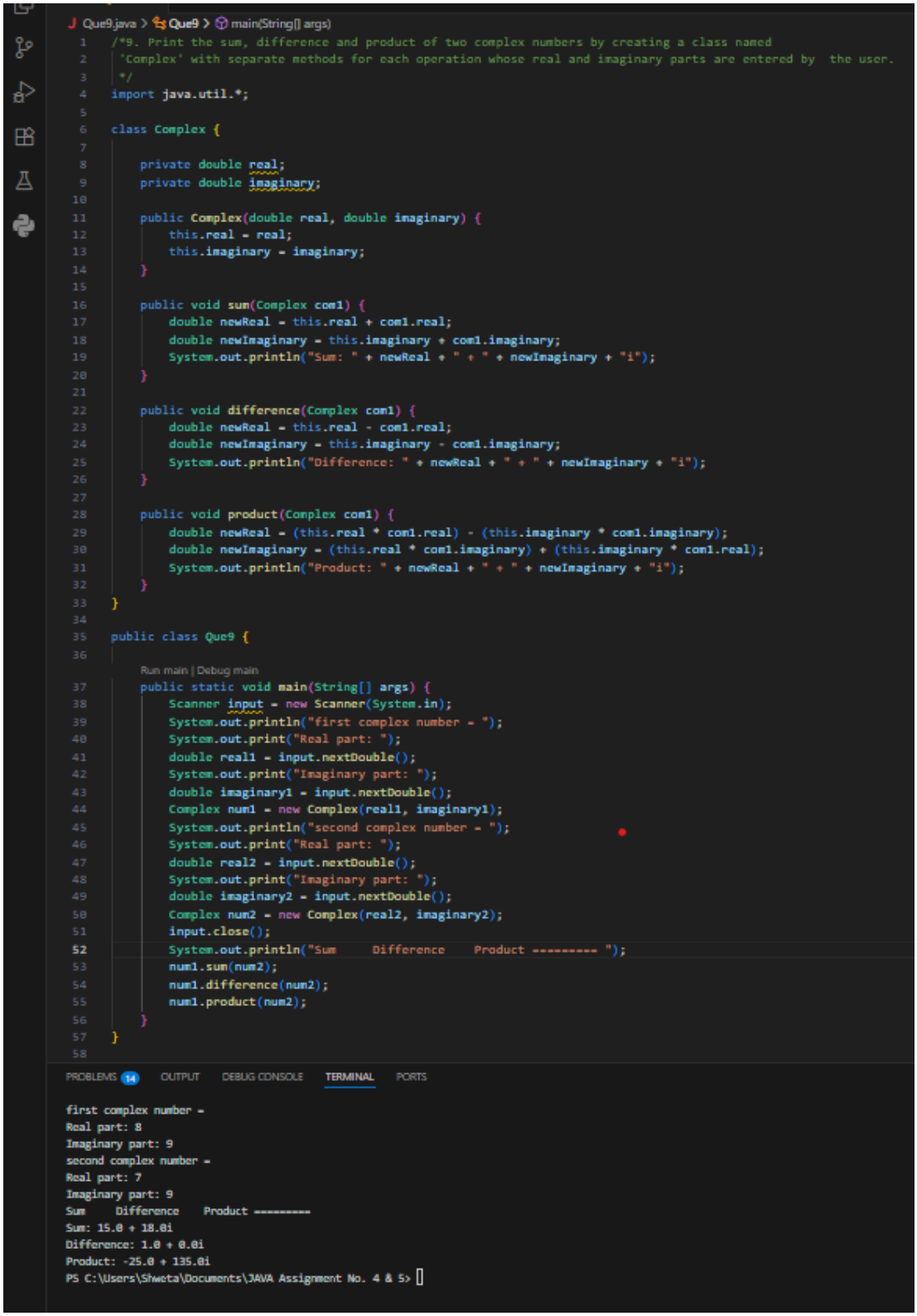
8. Print the average of three numbers entered by the user by creating a class named 'Average' having a method to calculate and print the average.



```java
//8. Print the average of three numbers entered by the user by creating a class named 'Average'  ha

import java.util.*;

class Average {

    public int num1;
    public int num2;
    public int num3;

    public Average(int p, int q, int r) {
        this.num1 = p;
        this.num2 = q;
        this.num3 = r;
    }

    public double calAvg() {
        double avg = (double) (num1 + num2 + num3) / 3;
        System.out.println("Average = " + avg);
        return avg;
    }
}

class Que8 {

    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Num1 =");
        int no1 = input.nextInt();
        System.out.println("Num2 =");
        int no2 = input.nextInt();
        System.out.println("Num3 =");
        int no3 = input.nextInt();
        input.close();
        Average avg = new Average(no1, no2, no3);
        avg.calAvg();

    }
}
```

```
PROBLEMS 11  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que8.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que8
Num1 =
8
Num2 =
9
Num3 =
9
Average = 8.666666666666666
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

9. Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by the user.



```java
/*9. Print the sum, difference and product of two complex numbers by creating a class named
 'Complex' with separate methods for each operation whose real and imaginary parts are entered by  the user.
 */
import java.util.*;

class Complex {

    private double real;
    private double imaginary;

    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }

    public void sum(Complex com1) {
        double newReal = this.real + com1.real;
        double newImaginary = this.imaginary + com1.imaginary;
        System.out.println("Sum: " + newReal + " + " + newImaginary + "i");
    }

    public void difference(Complex com1) {
        double newReal = this.real - com1.real;
        double newImaginary = this.imaginary - com1.imaginary;
        System.out.println("Difference: " + newReal + " + " + newImaginary + "i");
    }

    public void product(Complex com1) {
        double newReal = (this.real * com1.real) - (this.imaginary * com1.imaginary);
        double newImaginary = (this.real * com1.imaginary) + (this.imaginary * com1.real);
        System.out.println("Product: " + newReal + " + " + newImaginary + "i");
    }
}

public class Que9 {

    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("first complex number = ");
        System.out.print("Real part: ");
        double real1 = input.nextDouble();
        System.out.print("Imaginary part: ");
        double imaginary1 = input.nextDouble();
        Complex num1 = new Complex(real1, imaginary1);
        System.out.println("second complex number = ");
        System.out.print("Real part: ");
        double real2 = input.nextDouble();
        System.out.print("Imaginary part: ");
        double imaginary2 = input.nextDouble();
        Complex num2 = new Complex(real2, imaginary2);
        input.close();
        System.out.println("Sum     Difference    Product --------- ");
        num1.sum(num2);
        num1.difference(num2);
        num1.product(num2);
    }
}
```

```
first complex number =
Real part: 8
Imaginary part: 9
second complex number =
Real part: 7
Imaginary part: 9
Sum     Difference    Product ---------
Sum: 15.0 + 18.0i
Difference: 1.0 + 0.0i
Product: -25.0 + 135.0i
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

10. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name Year of joining Address

Robert 1994 64C- WallsStreat

Sam 2000 68D- WallsStreat

John 1999 26B- WallsStreat

```
 */
class Employee {

    private String name;
    private int yearOfJoining;
    private String address;

    public Employee(String n, int yoj, String a) {
        this.name = n;
        this.yearOfJoining = yoj;
        this.address = a;
    }

    public void display() {
        //System.out.println(this.name + "       " + this.yearOfJoining + "           " + this.address);
        System.out.printf("%-10s%-16d%s\n", this.name, this.yearOfJoining, this.address);
    }
}

class Que10 {

    Run main | Debug main
    public static void main(String[] args) {
        Employee robert = new Employee("Robert", 1994, "64C- WallsStreat");
        Employee sam = new Employee("Sam", 2000, "68D- WallsStreat");
        Employee john = new Employee("John", 1999, "26B- WallsStreat");

        System.out.println("Name     Year of joining       Address");
        robert.display();
        sam.display();
        john.display();
    }
}

PROBLEMS 18   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> javac Que10.java
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5> java Que10
Name     Year of joining       Address
Robert    1994           64C- WallsStreat
Sam       2000           68D- WallsStreat
John      1999           26B- WallsStreat
PS C:\Users\Shweta\Documents\JAVA Assignment No. 4 & 5>
```

11. Write a Java class Clock for dealing with the day time represented by hours, minutes, and seconds. Your class must have the following features:

- Three instance variables for the hours (range 0 - 23), minutes (range 0 - 59), and seconds (range 0 - 59).
- Three constructors:

  default (with no parameters passed; is should initialize the represented time to 12:0:0)

  a constructor with three parameters: hours, minutes, and seconds.

  a constructor with one parameter: the value of time in seconds since midnight (it should be converted into the time value in hours, minutes, and seconds)
- Instance methods:

  a *set*-method method setClock() with one parameter *seconds* since midnight (to be converted into the time value in hours, minutes, and seconds as above).

  get-methods getHours(), getMinutes(), getSeconds() with no parameters that return the corresponding values.

  set-methods setHours(), setMinutes(), setSeconds() with one parameter each that set up the corresponding instance variables.

  method tick() with no parameters that increments the time stored in a Clock object by one second.

  method addClock() accepting an object of type Clock as a parameter. The method should add the time represented by the parameter class to the time represented in the current class.

  Add an instance method toString() with no parameters to your class. toString() must return a String representation of the Clock object in the form "(hh:mm:ss)", for example "(03:02:34)".

  Add an instance method tickDown() which decrements the time stored in a Clock object by one second.

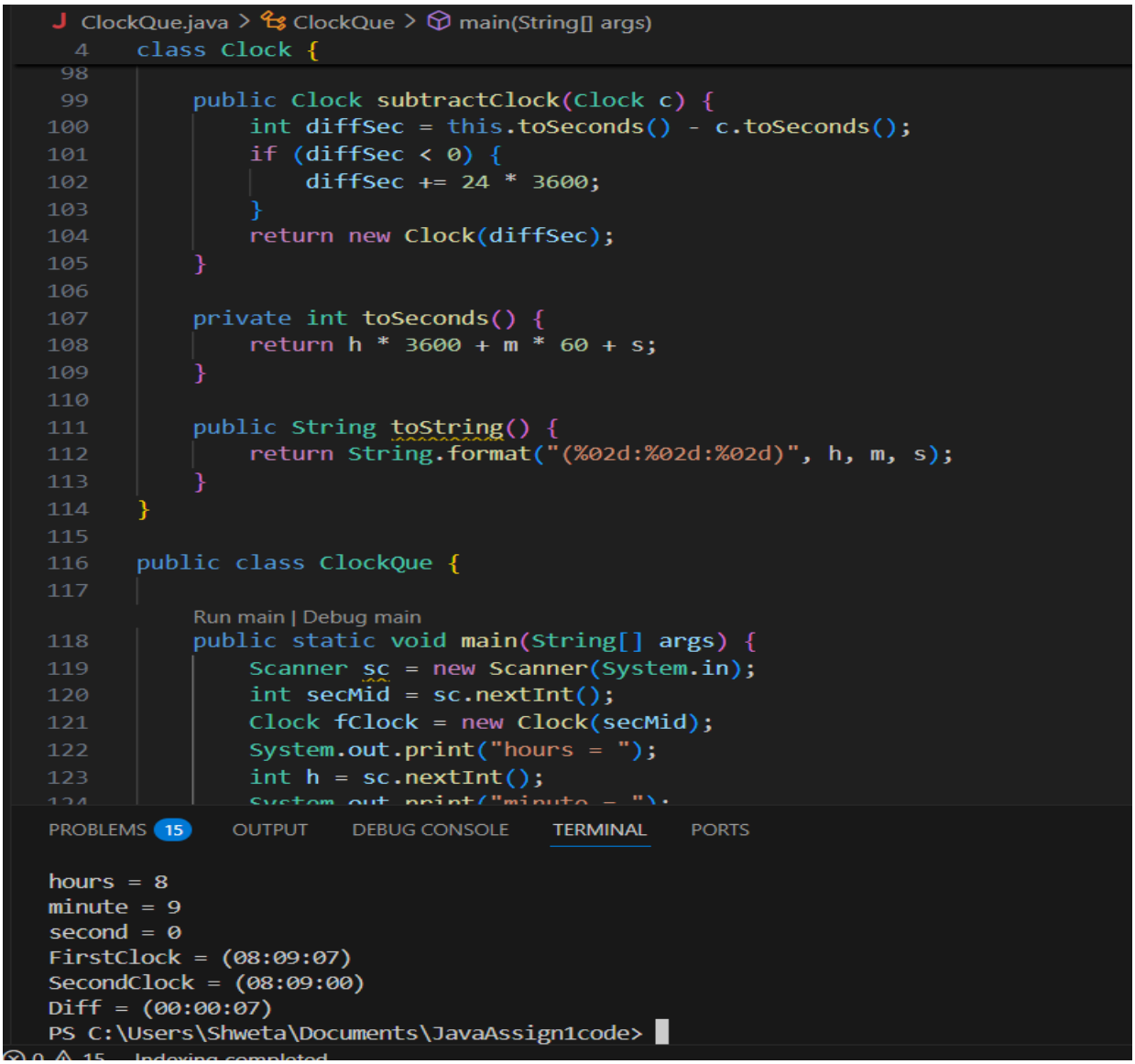
  Add an instance method subtractClock() that takes one Clock parameter and returns the difference between the time represented in the current Clock object and the one represented by the Clock parameter.

  Difference of time should be returned as a clock object.

Write a separate class ClockDemo with a main() method. The program should:

- instantiate a Clock object firstClock using one integer *seconds* since midnight obtained from the keyboard.
- tick the clock ten times by applying its *tick()* method and print out the time after each tick.
- Extend your code by appending to it instructions instantiating a Clock object secondClock by using three integers (hours, minutes, seconds) read from the keyboard.
- Then tick the clock ten times, printing the time after each tick.
- Add the secondClock time in firstClock by calling the method addClock. ● Print both clock objects calling toString method

Create a reference thirdClock that should reference the object of difference of firstClock and secondClock by calling the method subtractClock().



Question 3

Write a Java class Complex for dealing with complex numbers. Your class must have the following features:

- Instance variables :

  realPart for the real part of type double

  imaginaryPart for imaginary part of type double.

- Constructor:

  public Complex (): A default constructor, it should initialize the number to 0, 0)

  public Complex (double realPart, double imaginaryPart): A constructor with parameters, it creates the complex object by setting the two fields to the passed values.

- Instance methods:

  public Complex add (Complex otherNumber): This method will find the sum of the current complex number and the passed complex number. The method returns a new Complex number which is the sum of the two.

  public Complex subtract (Complex otherNumber): This method will find the difference of the current complex number and the passed complex number. The method returns a new Complex number which is the difference of the two.

  public Complex multiply (Complex otherNumber): This method will find the product of the current complex number and the passed complex number. The method returns a new Complex number which is the product of the two.

  public Complex divide (Complex otherNumber): This method will find the ... of the current complex number and the passed complex number. The method returns a new Complex number which is the ... of the two.

  public void setRealPart (double realPart): Used to set the real part of this complex number.

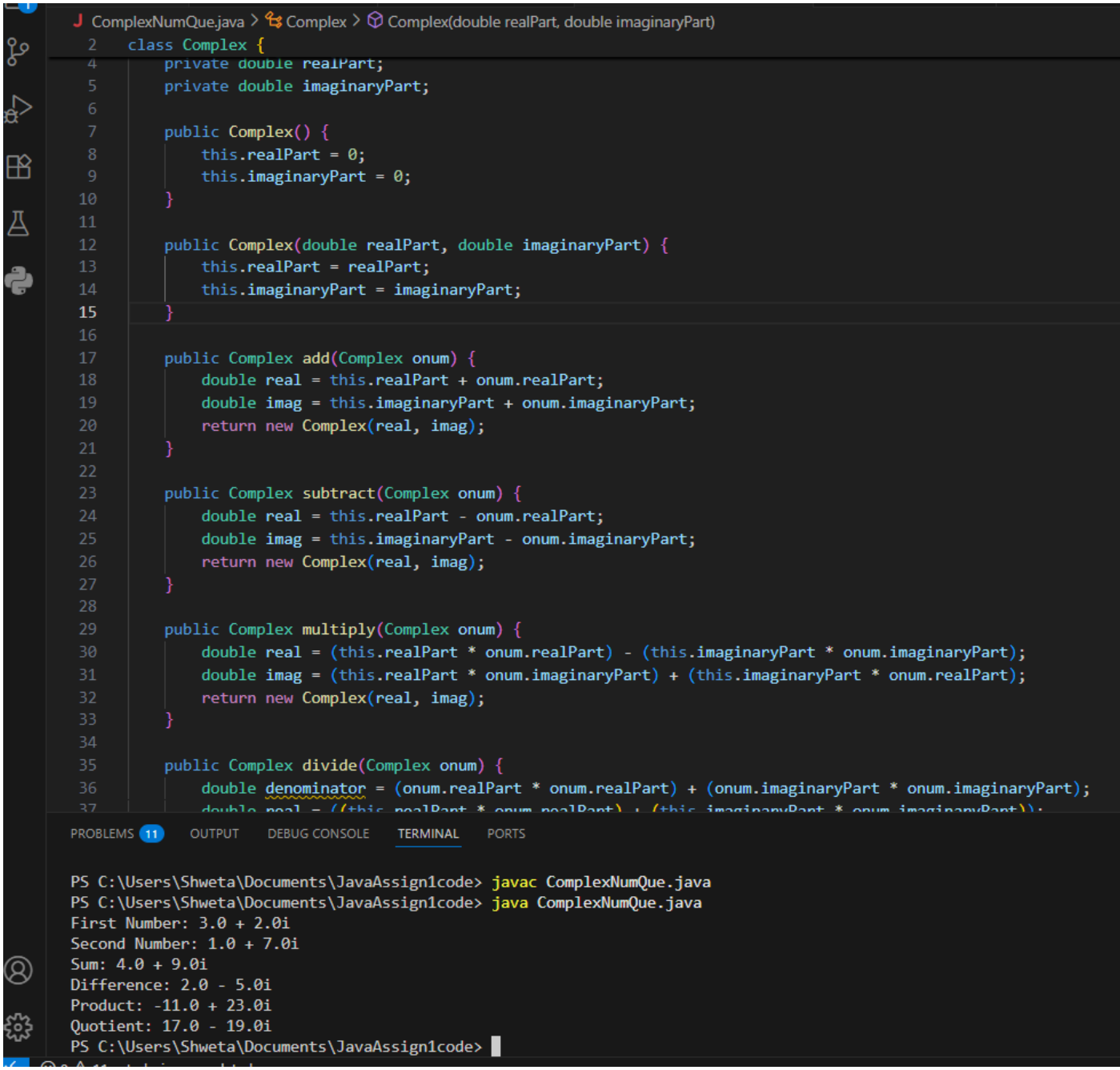
  public void setImaginaryPart (double realPart): Used to set the imaginary part of this complex number.

  public double getRealPart(): This method returns the real part of the complex number

  public double getImaginaryPart(): This method returns the imaginary part of the complex number

  public String toString(): This method allows the complex number to be easily printed out to the screen

Write a separate class **ComplexDemo** with a main() method and test the Complex class methods.



Question 4

Write a Java class Author with following features:

- Instance variables :

  firstName for the author's first name of type String.

  lastName for the author's last name of type String.

- Constructor:

  public Author (String firstName, String lastName): A constructor with

parameters, it creates the Author object by setting the two fields to the passed values.

- Instance methods:

  public void setFirstName (String firstName): Used to set the first name of the author.

  public void setLastName (String lastName): Used to set the last name of the author.

  public double getFirstName(): This method returns the first name of the author.

  public double getLastName(): This method returns the last name of the author.

  public String toString(): This method printed out author's name to the screen

```java
class Author {

    String firstName;
    String lastName;

    public Author(String fn, String ln) {
        this.firstName = fn;
        this.lastName = ln;
    }

    public void setFirstName(String fn) {
        this.firstName = fn;
    }

    public void setLastName(String ln) {
        this.lastName = ln;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String toString() {
        return firstName + " " + lastName;
    }
}
class AuthorQue {
    public static void main(String[] args) {
        Author a = new Author("Maharshi", "Valmiki");
        System.out.println("Author = " + a.toString());
        a.setFirstName("Tulsi");
        a.setLastName("Das");
        System.out.println("First Name = " + a.getFirstName());
        System.out.println("Last Name = " + a.getLastName());
    }
}
```

```
PROBLEMS 29   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> javac AuthorQue.java
PS C:\Users\Shweta\Documents\JavaAssign1code> java AuthorQue
Author = Maharshi Valmiki
First Name = Tulsi
Last Name = Das
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

Write a Java class Book with following features:

- Instance variables :

  title for the title of book of type String.

  author for the author's name of type String.

  price for the book price of type double.

- Constructor:

  public Book (String title, Author name, double price): A constructor with parameters, it creates the Author object by setting the fields to the passed values.

- Instance methods:

  public void setTitle(String title): Used to set the title of a book.

  public void setAuthor(String author): Used to set the name of the author of a book.

  public void setPrice(double price): Used to set the price of a book.

  public double getTitle(): This method returns the title of the book.

  public double getAuthor(): This method returns the author's name of the book.

  public String toString(): This method printed out book's details to the screen

Write a separate class **BookDemo** with a main() method creates a Book titled “Developing Java Software” with authors Russel Winderand price 79.75. Prints the Book’s string representation to standard output (using System.out.println).

1.Write a program by creating an 'Employee' class having the following methods and print the final salary.

1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as

parameter2 - 'AddSal()' which adds $10 to the salary of the employee if it is less than $500.

3 - 'AddWork()' which adds $5 to the salary of an employee if the number of hours of work per day is more than 6 hours.

```
 */
class Employee {
    double sal;
    int hrs;
    void getInfo(double s, int h) {
        this.sal = s;
        this.hrs = h;
    }
    void addSal() {
        if (this.sal < 500) {
            this.sal += 10;
        }
    }
    void addWork() {
        if (this.hrs > 6) {
            this.sal += 5;
        }
    }
    void printSal() {
        System.out.println("Total Salary = " + this.sal);
    }
}
class EmployeeSal {

    Run main | Debug main
    public static void main(String[] args) {
        Employee emp = new Employee();
        emp.getInfo(900.0, 7);
        emp.addSal();
        emp.addWork();
        emp.printSal();
        Employee emp1 = new Employee();
        emp1.getInfo(200.0, 4);
        emp1.addSal();
        emp1.addWork();
        emp1.printSal();
    }
}
```

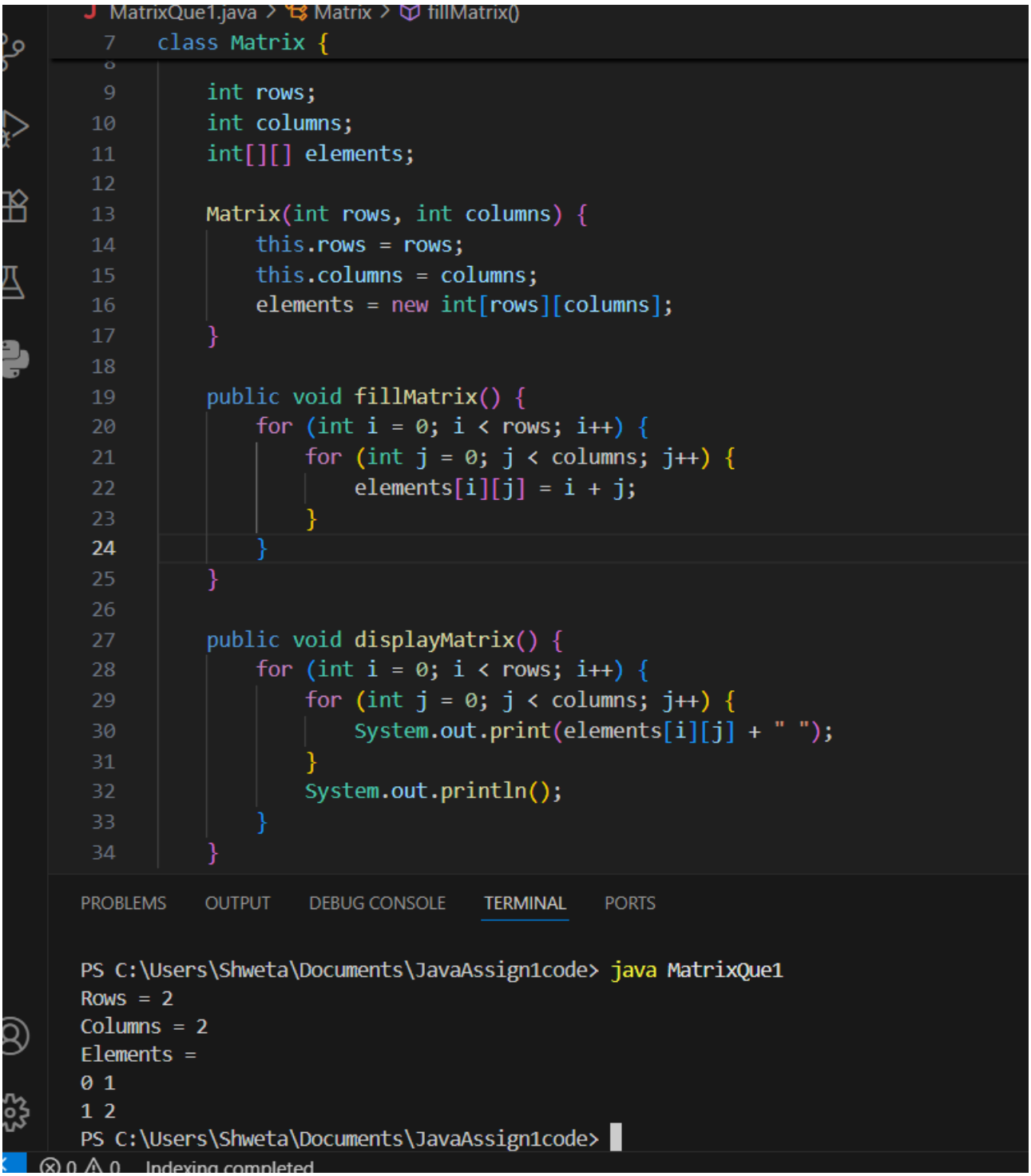
```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> javac EmployeeSal.java
PS C:\Users\Shweta\Documents\JavaAssign1code> java EmployeeSal
Total Salary = 905.0
Total Salary = 210.0
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

2. Create a class called 'Matrix' containing a constructor that initializes the number of rows and number of columns of a new Matrix object. The Matrix class has the following information:

1 - number of rows of matrix

2 - number of columns of matrix

3 - elements of matrix in the form of 2D array



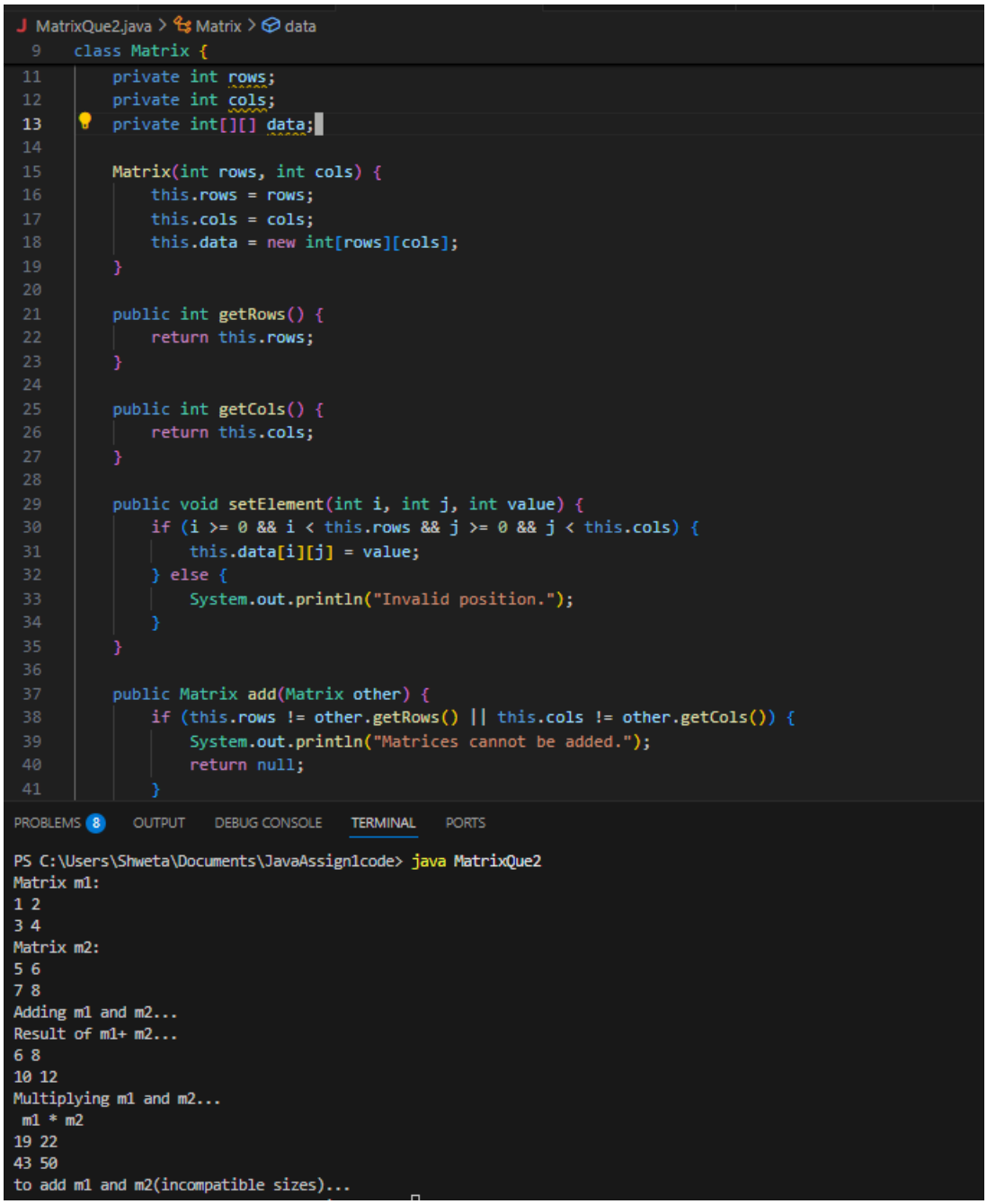
3.The Matrix class has methods for each of the following:

1 - get the number of rows
2 - get the number of columns

3 - set the elements of the matrix at given position (i,j)

4 - adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed.

5 - multiplying the two matrices



```
MatrixQue2.java > Matrix > data
 9  class Matrix {
11      private int rows;
12      private int cols;
13      private int[][] data;
14
15      Matrix(int rows, int cols) {
16          this.rows = rows;
17          this.cols = cols;
18          this.data = new int[rows][cols];
19      }
20
21      public int getRows() {
22          return this.rows;
23      }
24
25      public int getCols() {
26          return this.cols;
27      }
28
29      public void setElement(int i, int j, int value) {
30          if (i >= 0 && i < this.rows && j >= 0 && j < this.cols) {
31              this.data[i][j] = value;
32          } else {
33              System.out.println("Invalid position.");
34          }
35      }
36
37      public Matrix add(Matrix other) {
38          if (this.rows != other.getRows() || this.cols != other.getCols()) {
39              System.out.println("Matrices cannot be added.");
40              return null;
41          }
```

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> java MatrixQue2
Matrix m1:
1 2
3 4
Matrix m2:
5 6
7 8
Adding m1 and m2...
Result of m1+ m2...
6 8
10 12
Multiplying m1 and m2...
 m1 * m2
19 22
43 50
to add m1 and m2(incompatible sizes)...
```

****

*1. Constructor Chaining within the Same Class

Create a class Car with multiple constructors that initialize different attributes using constructor chaining.

Problem Statement:

- Create a class Car with attributes brand, model, and price.
- Implement **constructor chaining** within the same class:
    - One constructor should only take the brand.
    - Another constructor should take brand and model.
    - The final constructor should take brand, model, and price.
- Use the this() keyword to call other constructors.
- Display car details in each constructor.

☑ **Task:** Create objects using different constructors and observe constructor chaining in action.

CarConstructorQue1.java > Car

```java
class Car {

    String brand;
    String model;
    double price;

    Car(String b) {
        this.brand = b;
        System.out.println("Brand =" + b);
    }

    Car(String b, String m) {
        this(b);
        this.model = m;
        System.out.println("Model =" + m);
    }

    Car(String b, String m, double p) {
        this(b, m);
        this.price = p;
        System.out.println("Price =" + p);
    }
}

public class CarConstructorQue1 {

    Run main | Debug main
    public static void main(String[] args) {
        Car c = new Car("BMW");
        Car c1 = new Car("Audi", "Q7");
        new Car("Rolls-Royce", "Phantom", 1000000);
    }
}
```

PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Shweta\Documents\JavaAssign1code> javac CarConstructorQue1.java
PS C:\Users\Shweta\Documents\JavaAssign1code> java CarConstructorQue1
Brand =BMW
Brand =Audi
Model =Q7
Brand =Rolls-Royce
Model =Phantom
Price =1000000.0
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

2. Constructor Chaining Using super Keyword (Parent-Child Relationship)

Create a class hierarchy where the child class calls the parent class constructor using super().

Problem Statement:

- Create a Person class with attributes name and age.
- Create a Student class that extends Person and has an additional attribute course. ● Use constructor chaining:
  - ○ Person class should have a constructor initializing name and age. ○ Student class should use super(name, age) to call the Person constructor and then initialize course.
- Display details in both constructors.

☑ **Task:** Create a Student object and verify that both constructors (parent and child) are executed in sequence.

```
10  class Person {
11
12      String name;
13      int age;
14
15      Person(String n, int a) {
16          this.name = n;
17          this.age = a;
18          System.out.println("Person: " + n + ",& Age = " + a);
19      }
20  }
21
22  class Student extends Person {
23
24      String course;
25
26      Student(String n, int a, String c) {
27          super(n, a);
28          this.course = c;
29          System.out.println("Course =" + c);
30      }
31  }
32
33  class PersonConstructorQue2 {
34
        Run main | Debug main
35      public static void main(String[] args) {
36          Student s = new Student("Abc", 21, "DBDA");
37      }
38
39  }
40
```

```
PROBLEMS 16   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> javac PersonConstructorQue2.java
PS C:\Users\Shweta\Documents\JavaAssign1code> java PersonConstructorQue2
Person: Abc,& Age = 21
Course =DBDA
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

3. Multi-Level Constructor Chaining (Grandparent → Parent → Child) Demonstrate constructor chaining in a multi-level inheritance scenario. Problem Statement:

- Create a Vehicle class with an attribute type.
  - Create a subclass FourWheeler with an additional attribute brand. ● Create another subclass Car with attributes model and price.
  - Use **multi-level constructor chaining**:
    - Vehicle initializes type.
    - FourWheeler calls super(type) and initializes brand.
    - Car calls super(type, brand), initializes model, and price.
  - Display details at each level.

☑ **Task:** Create a Car object and verify that constructors are executed from parent → child → grandchild.

```java
class Vehicle {

    String t;

    Vehicle(String t) {
        this.t = t;
        System.out.println("Vehicle =" + t);
    }
}

class FourWheeler extends Vehicle {

    String b;

    FourWheeler(String t, String b) {
        super(t);
        this.b = b;
        System.out.println("Four Wheeler Brand =" + b);
    }
}

class Car extends FourWheeler {

    String m;
    double p;

    Car(String t, String b, String m, double p) {
        super(t, b);
        this.m = m;
        this.p = p;
        System.out.println("Car Model =" + m);
        System.out.println("Car Price =" + p);
    }
}

public class VehicalConstructorQue3 {

    Run main | Debug main
    public static void main(String[] args) {
        Car c = new Car("Four Wheeler", "Audi", "A4", 9900000.);
    }
}
```

```
PROBLEMS 11   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> java VehicalConstructorQue3
Vehicle =Four Wheeler
Four Wheeler Brand =Audi
Car Model =A4
Car Price =9900000.0
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

4. Constructor Chaining in an E-Commerce Scenario

Create an Order class where different constructors initialize order details using constructor chaining.

Problem Statement:

- Create an Order class with attributes orderId, customerName, and totalAmount.
- Implement **constructor chaining** within the same class:
  - One constructor initializes only orderId.
  - Another constructor initializes orderId and customerName by calling the first constructor.
  - The final constructor initializes all three attributes (orderId, customerName, totalAmount) by calling the second constructor.
- Display order details.

☑ **Task:** Create Order objects using different constructors and verify how chaining works.

```java
class Order {

    int orderId;
    String customerName;
    double totalAmount;

    Order(int o) {
        this.orderId = o;
        System.out.println("Order ID =" + o);
    }

    Order(int o, String c) {
        this(o);
        this.customerName = c;
        System.out.println("Customer Name =" + c);
    }

    Order(int o, String c, double t) {
        this(o, c);
        this.totalAmount = t;
        System.out.println("Total Amount =" + t);
    }
}

class OrderConstructorQue4 {

    Run main | Debug main
    public static void main(String[] args) {
        Order o = new Order(101);
        Order o1 = new Order(102, "aaa");
        Order o2 = new Order(103, "bbb", 1000);
    }
}
```

```
PROBLEMS 23   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Shweta\Documents\JavaAssign1code> java OrderConstructorQue4
Order ID =101
Order ID =102
Customer Name =aaa
Order ID =103
Customer Name =bbb
Total Amount =1000.0
PS C:\Users\Shweta\Documents\JavaAssign1code>
```

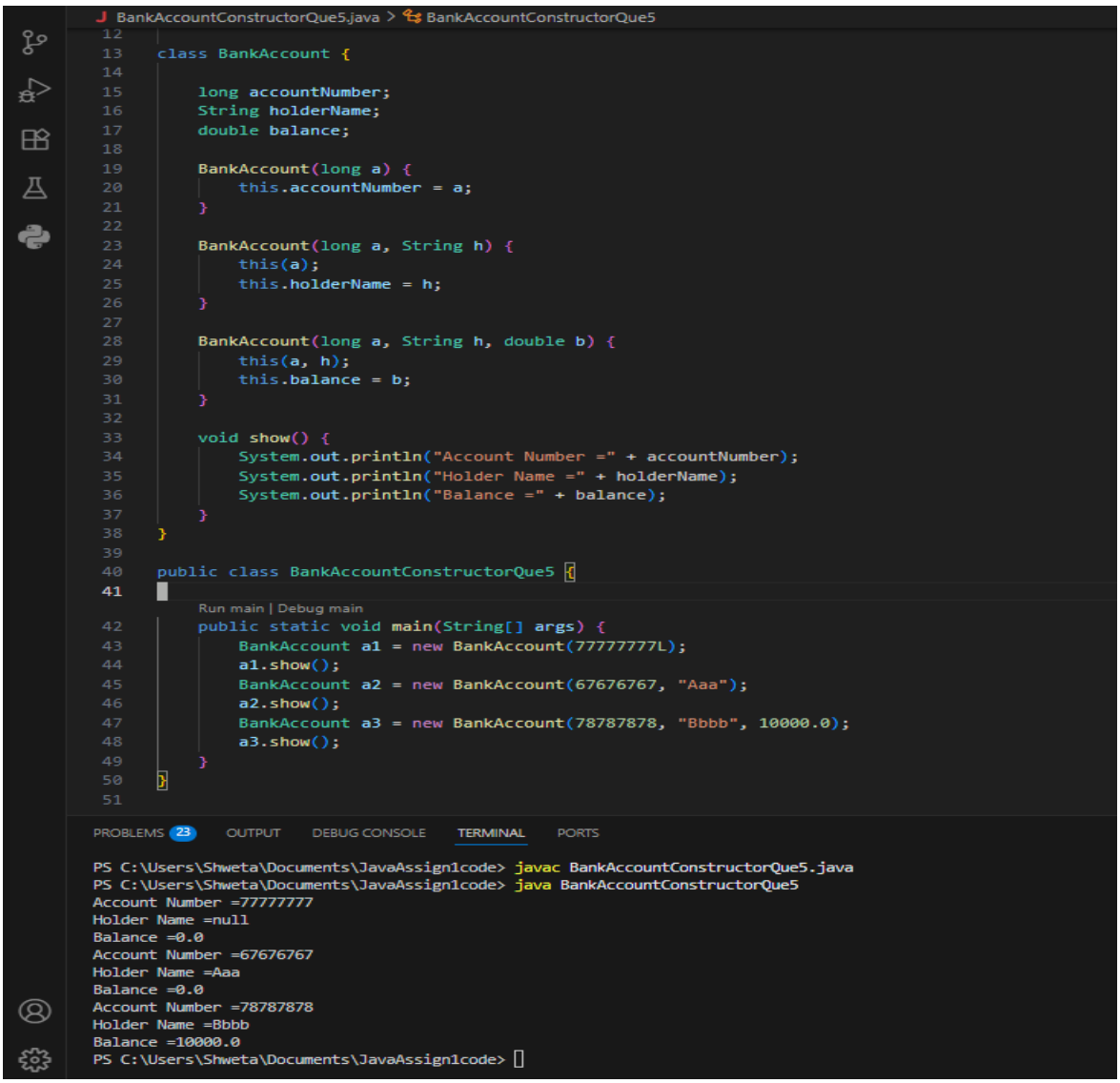
5. Constructor Chaining in a Banking System

Create a `BankAccount` class where constructor chaining initializes different account types.

Problem Statement:

- Create a `BankAccount` class with attributes `accountNumber`, `holderName`, and `balance`.
- Implement **constructor chaining**:
  - One constructor initializes only `accountNumber`.
  - Another constructor initializes `accountNumber` and `holderName`, calling the first constructor.
  - The final constructor initializes all three attributes by calling the second constructor.
- Implement a method to **display account details**.

☑ **Task:** Create `BankAccount` objects using different constructors and verify the constructor call sequence.


BankAccountConstructorQue5.java > BankAccountConstructorQue5
class BankAccount {
long accountNumber;
String holderName;
double balance;
BankAccount(long a) {
this.accountNumber = a;
}
BankAccount(long a, String h) {
this(a);
this.holderName = h;
}
BankAccount(long a, String h, double b) {
this(a, h);
this.balance = b;
}
void show() {
System.out.println("Account Number =" + accountNumber);
System.out.println("Holder Name =" + holderName);
System.out.println("Balance =" + balance);
}
}
public class BankAccountConstructorQue5 {
Run main | Debug main
public static void main(String[] args) {
BankAccount a1 = new BankAccount(77777777L);
a1.show();
BankAccount a2 = new BankAccount(67676767, "Aaa");
a2.show();
BankAccount a3 = new BankAccount(78787878, "Bbbb", 10000.0);
a3.show();
}
}
PROBLEMS 23
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Shweta\Documents\JavaAssign1code> javac BankAccountConstructorQue5.java
PS C:\Users\Shweta\Documents\JavaAssign1code> java BankAccountConstructorQue5
Account Number =77777777
Holder Name =null
Balance =0.0
Account Number =67676767
Holder Name =Aaa
Balance =0.0
Account Number =78787878
Holder Name =Bbbb
Balance =10000.0
PS C:\Users\Shweta\Documents\JavaAssign1code>