School of Computer Science and Engineering (SCOPE)

Digital Assignment

Fall Semester 2024-25

COURSE CODE: BCSE103E

COURSE TITLE: Computer Programming: Java

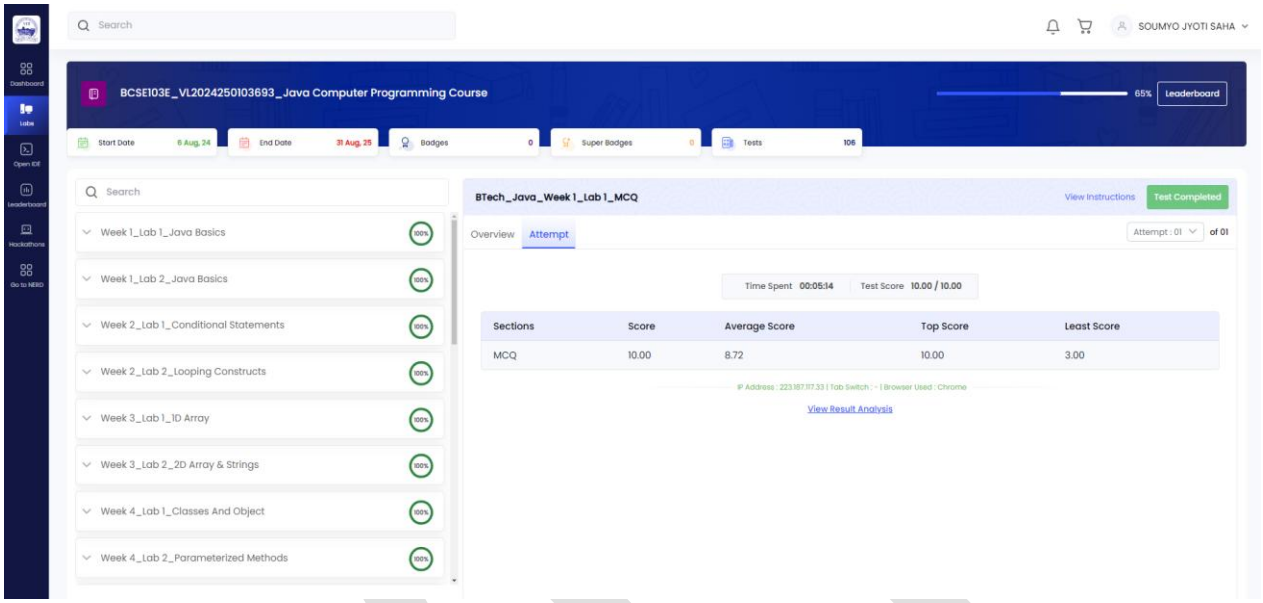
Name: Soumyojyoti Saha

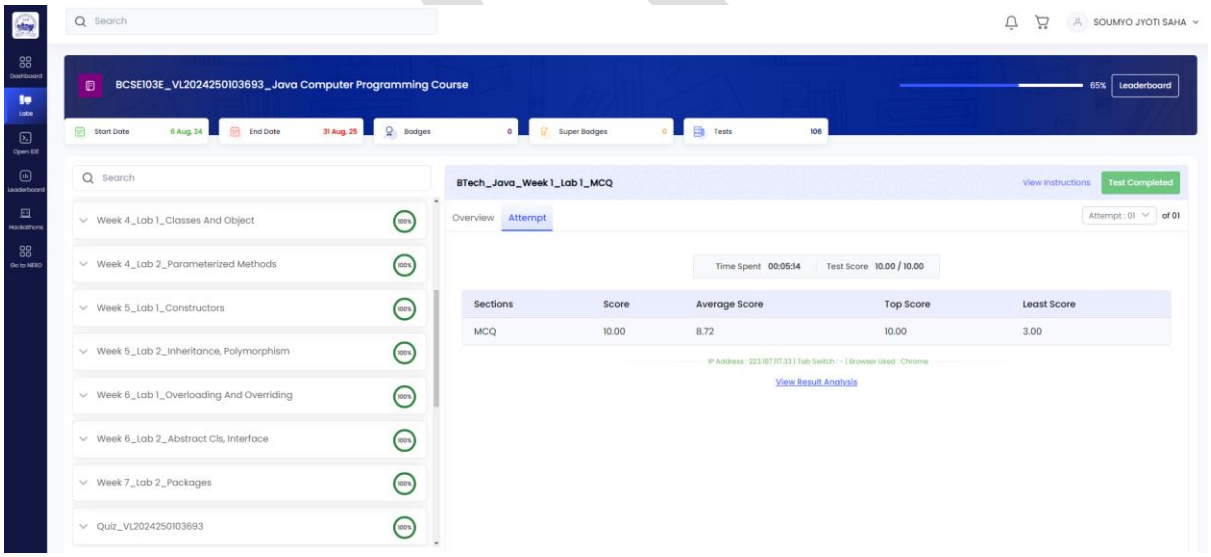
Register Number: 21BCE4007

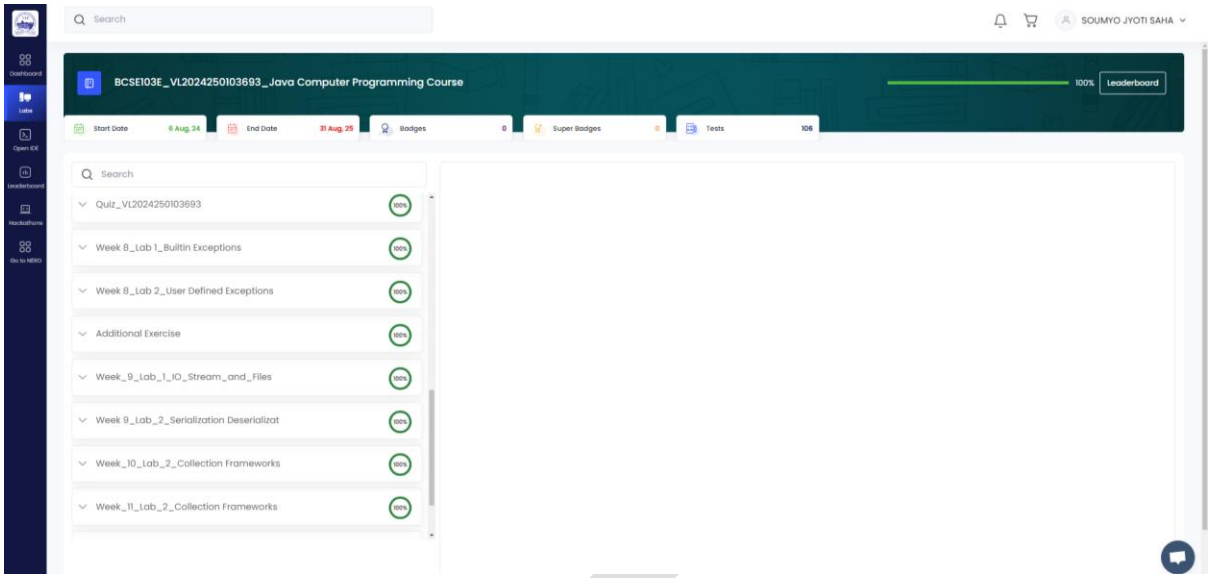
JAVA LAB EXERCISES

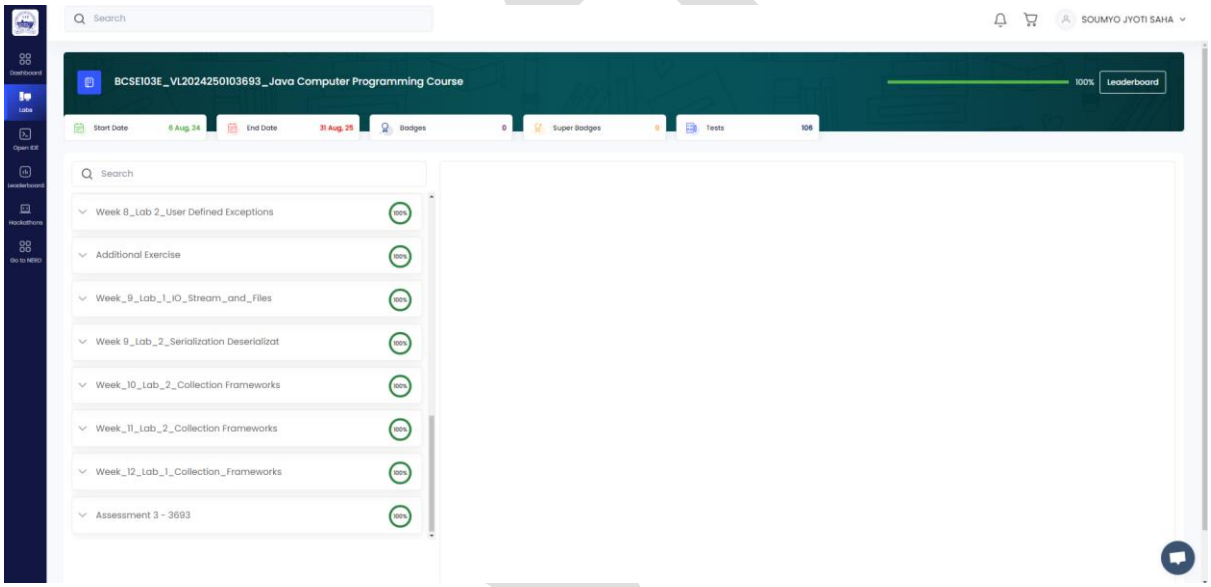
NEOCODELAB WEEK ASSESSMENT COMPLETION STATUS

#Screen shot of your week 1 to week 9 or 12 Completion Status






Search
SOUMYO JYOTI SAHA
Dashboard
Labs
Open IDE
Leaderboard
Hackathons
Go to NEXT
BCSE103E_VL2024250103693_Java Computer Programming Course
100%
Leaderboard
Start Date
6 Aug, 24
End Date
31 Aug, 25
Badges
0
Super Badges
0
Tests
106
Search
Quiz_VL2024250103693
Week 8_Lab 1_Builtin Exceptions
Week 8_Lab 2_User Defined Exceptions
Additional Exercise
Week_9_Lab_1_IO_Stream_and_Files
Week 9_Lab_2_Serialization Deserializat
Week_10_Lab_2_Collection Frameworks
Week_11_Lab_2_Collection Frameworks


Search
SOUMYO JYOTI SAHA
Dashboard
Labs
Open IDE
Leaderboard
Hackathons
Go to NEXT
BCSE103E_VL2024250103693_Java Computer Programming Course
100%
Leaderboard
Start Date
6 Aug, 24
End Date
31 Aug, 25
Badges
0
Super Badges
0
Tests
106
Search
Week 8_Lab 2_User Defined Exceptions
Additional Exercise
Week_9_Lab_1_IO_Stream_and_Files
Week 9_Lab_2_Serialization Deserializat
Week_10_Lab_2_Collection Frameworks
Week_11_Lab_2_Collection Frameworks
Week_12_Lab_1_Collection_Frameworks
Assessment 3 - 3693

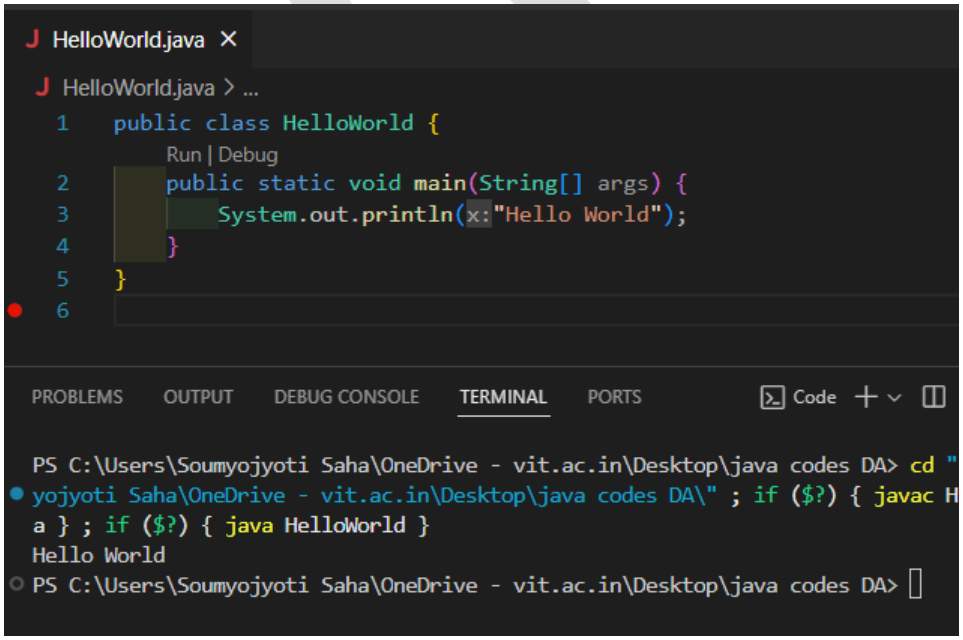
1. Write a Java Program to print “Hello World”.

Aim: To write a Java program that prints “Hello World” to the console.

Source Code:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Input / Output:



2. Write a Java Program to find the greatest among the three numbers.

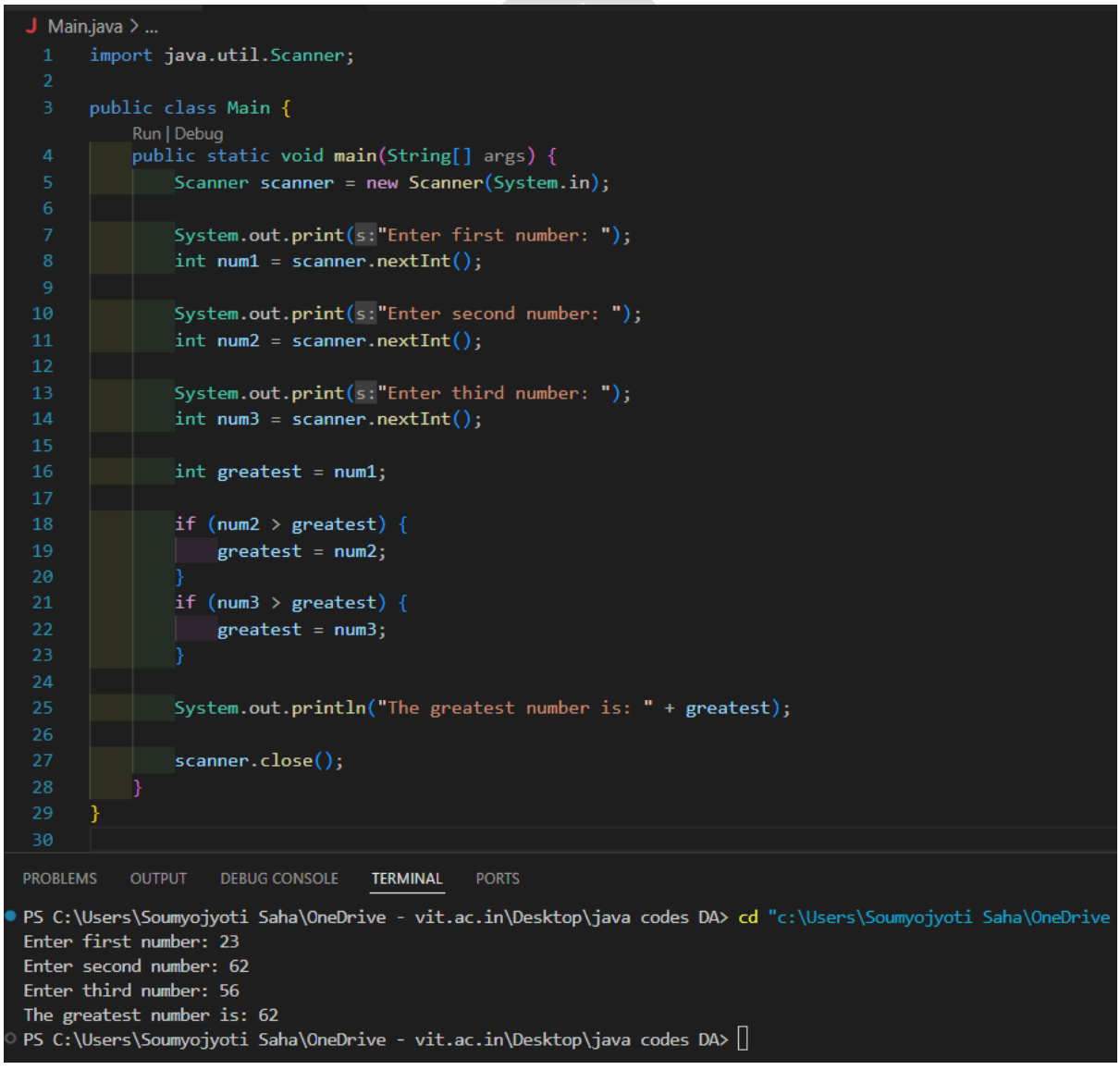
Aim: To write a Java program that finds and prints the greatest of three given numbers.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter first number: ");
        int num1 = scanner.nextInt();


        System.out.print("Enter second number: ");
        int num2 = scanner.nextInt();


        System.out.print("Enter third number: ");
        int num3 = scanner.nextInt();


        int greatest = num1;


        if (num2 > greatest) {
            greatest = num2;
        }
        if (num3 > greatest) {
            greatest = num3;
```

```
        }

        System.out.println("The greatest number is: " + greatest);

        scanner.close();
    }
}
```

Input / Output:



3. Write a Java program to print the addition, multiply, subtract, divide and reminder of two numbers.

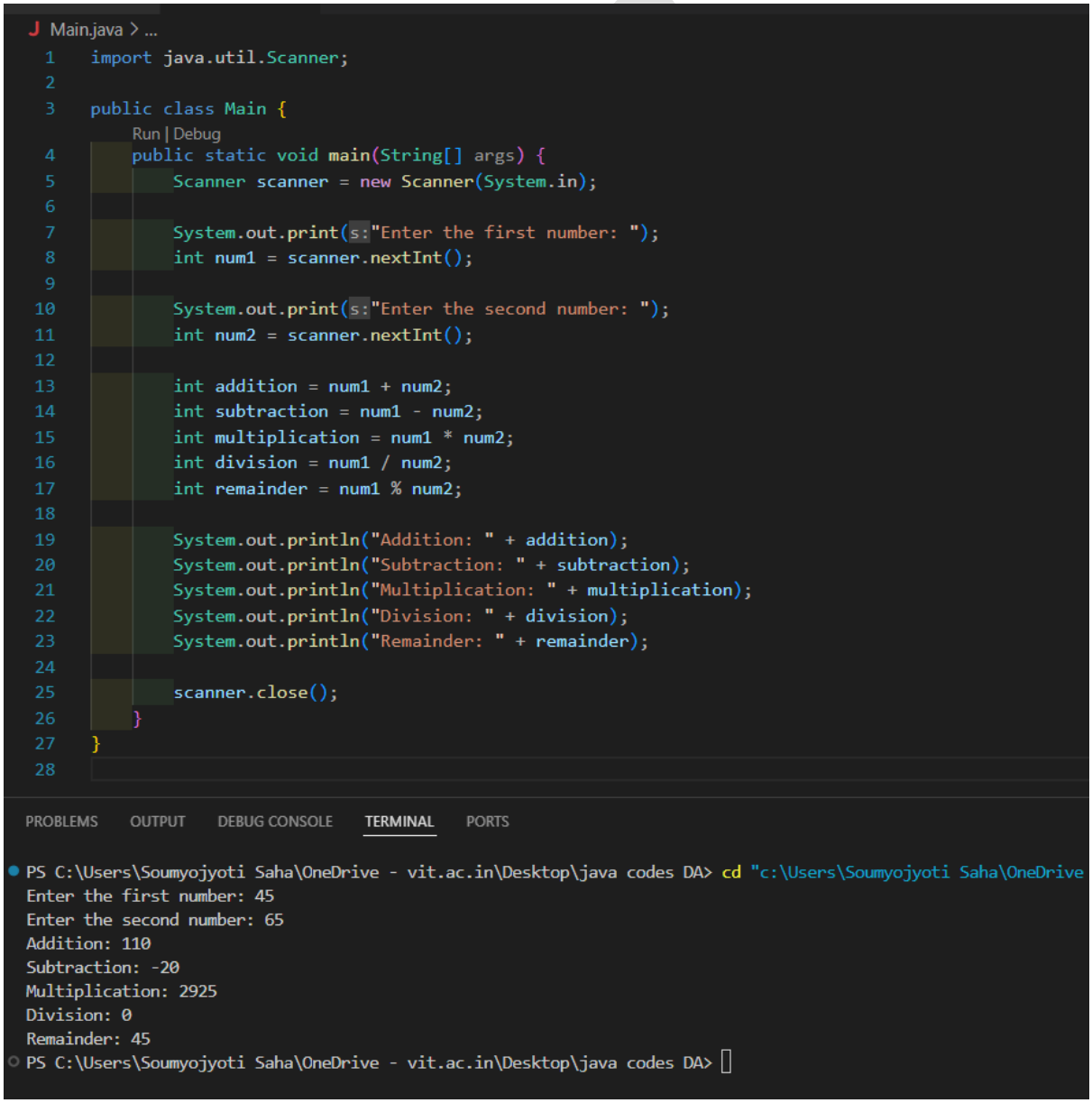
Aim: To write a Java program that performs and prints the results of addition, subtraction, multiplication, division, and remainder operations on two given numbers.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();


        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();


        int addition = num1 + num2;
        int subtraction = num1 - num2;
        int multiplication = num1 * num2;
        int division = num1 / num2;
        int remainder = num1 % num2;


        System.out.println("Addition: " + addition);
        System.out.println("Subtraction: " + subtraction);
        System.out.println("Multiplication: " + multiplication);
        System.out.println("Division: " + division);
```

```
        System.out.println("Remainder: " + remainder);


        scanner.close();
    }
}
```

Input / Output:



```
Main.java > ...
1   import java.util.Scanner;
2
3   public class Main {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6
7           System.out.print(s:"Enter the first number: ");
8           int num1 = scanner.nextInt();
9
10          System.out.print(s:"Enter the second number: ");
11          int num2 = scanner.nextInt();
12
13          int addition = num1 + num2;
14          int subtraction = num1 - num2;
15          int multiplication = num1 * num2;
16          int division = num1 / num2;
17          int remainder = num1 % num2;
18
19          System.out.println("Addition: " + addition);
20          System.out.println("Subtraction: " + subtraction);
21          System.out.println("Multiplication: " + multiplication);
22          System.out.println("Division: " + division);
23          System.out.println("Remainder: " + remainder);
24
25          scanner.close();
26      }
27  }
28

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the first number: 45
Enter the second number: 65
Addition: 110
Subtraction: -20
Multiplication: 2925
Division: 0
Remainder: 45
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

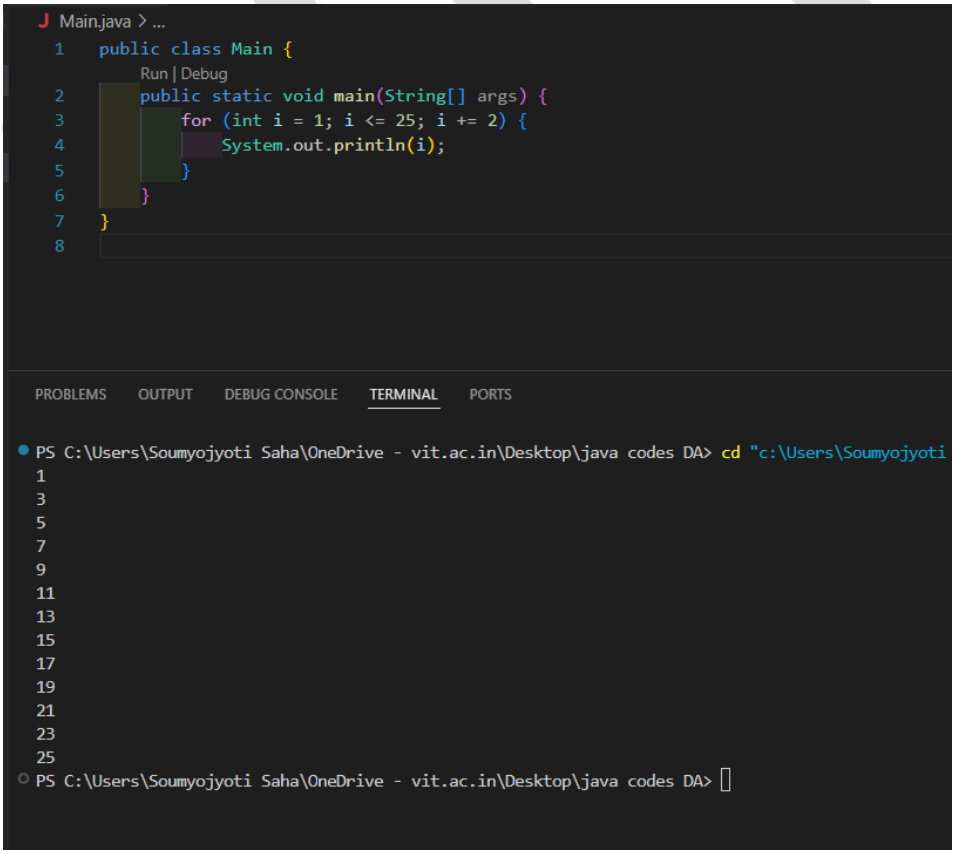
4. Write a Java program to print all the odd numbers from 1-25. Print one number per line.

Aim: To write a Java program that prints all odd numbers from 1 to 25, with each number displayed on a new line.

Source Code:

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 25; i += 2) {
            System.out.println(i);
        }
    }
}
```

Input / Output:



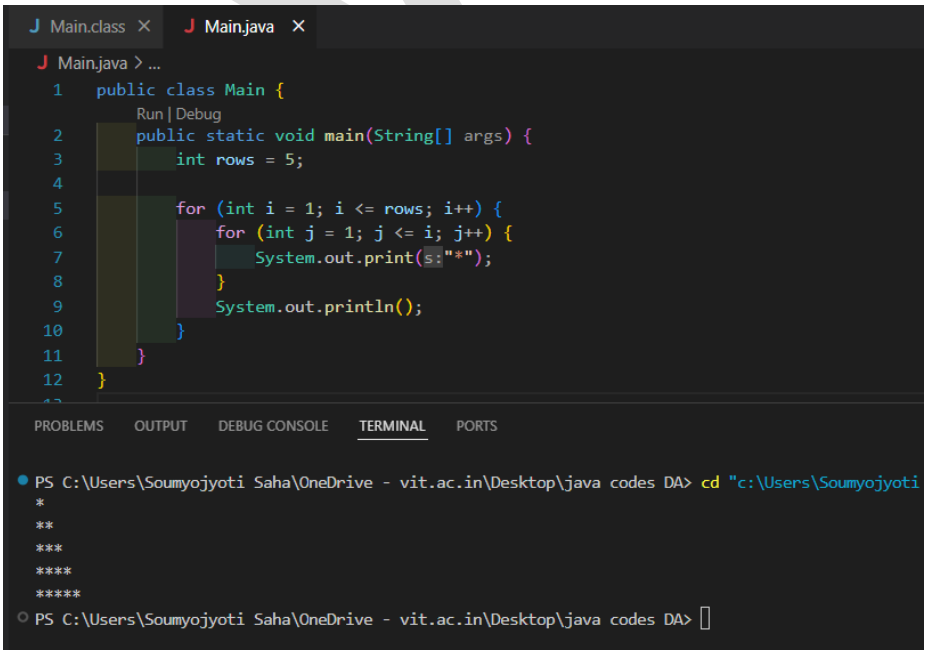
5. Write a Java program to print right angled * pattern.

Aim: To write a Java program that prints a right-angled triangle pattern using asterisks (*), with each row displaying one additional asterisk up to a specified number of rows (5 in this case).

Source Code:

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Input / Output:



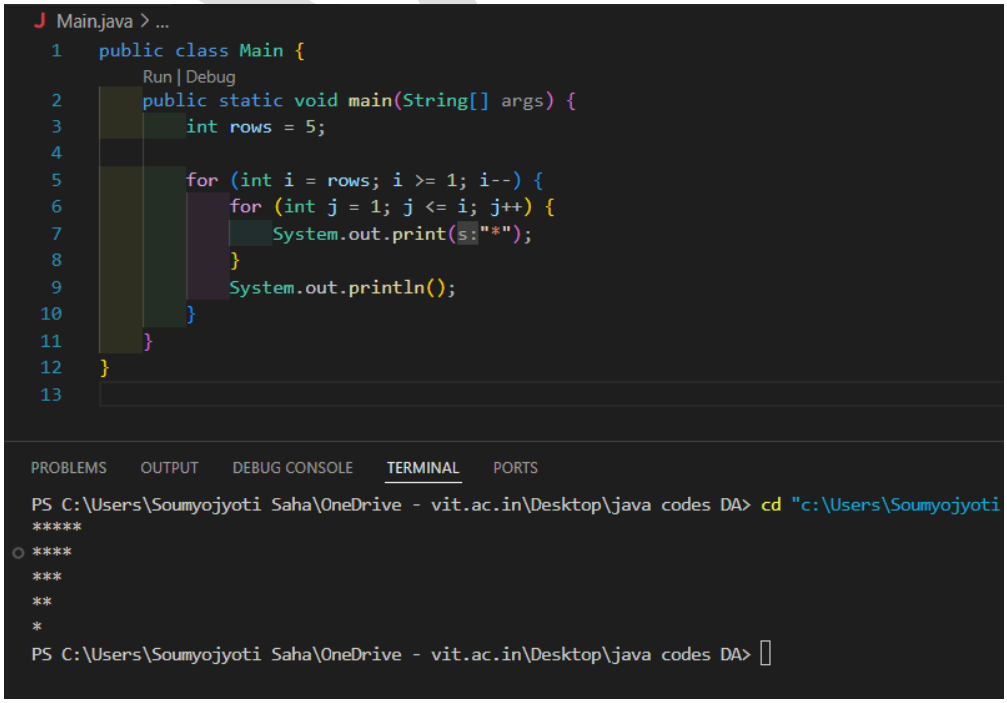
6. Write a Java Program to print Inverted Right triangle Pattern.

Aim: To write a Java program that prints an inverted right-angled triangle pattern using asterisks (*), with each row displaying one less asterisk until only one remains.

Source Code:

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;


        for (int i = rows; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Input / Output:



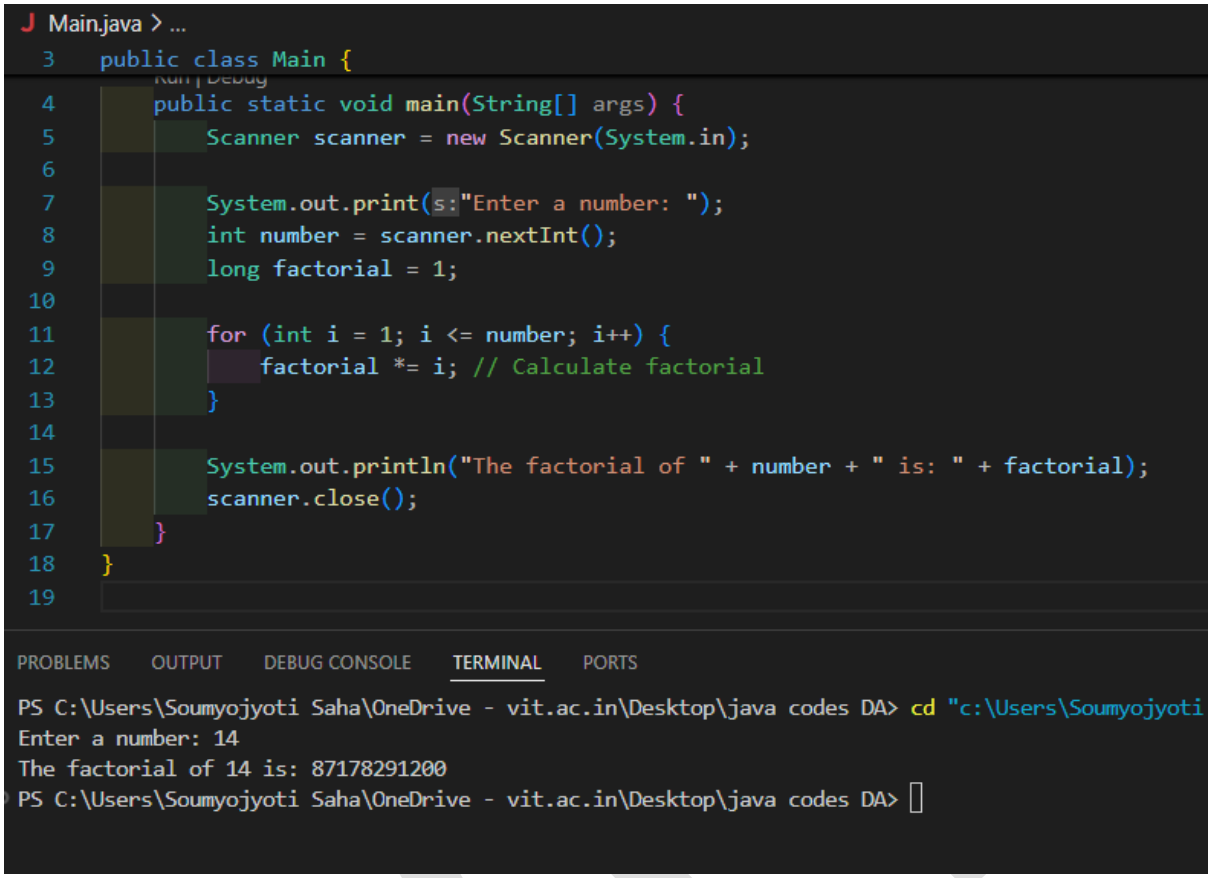
7. Write a Java Program to print the Factorial of a number.

Aim: To write a Java program that calculates and prints the factorial of a given number by multiplying all positive integers up to that number.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        long factorial = 1;


        for (int i = 1; i <= number; i++) {
            factorial *= i; // Calculate factorial
        }


        System.out.println("The factorial of " + number + " is: " + factorial);
        scanner.close();
    }
}
```

Input / Output:



8. Write a Java Program to print the Fibonacci Series.

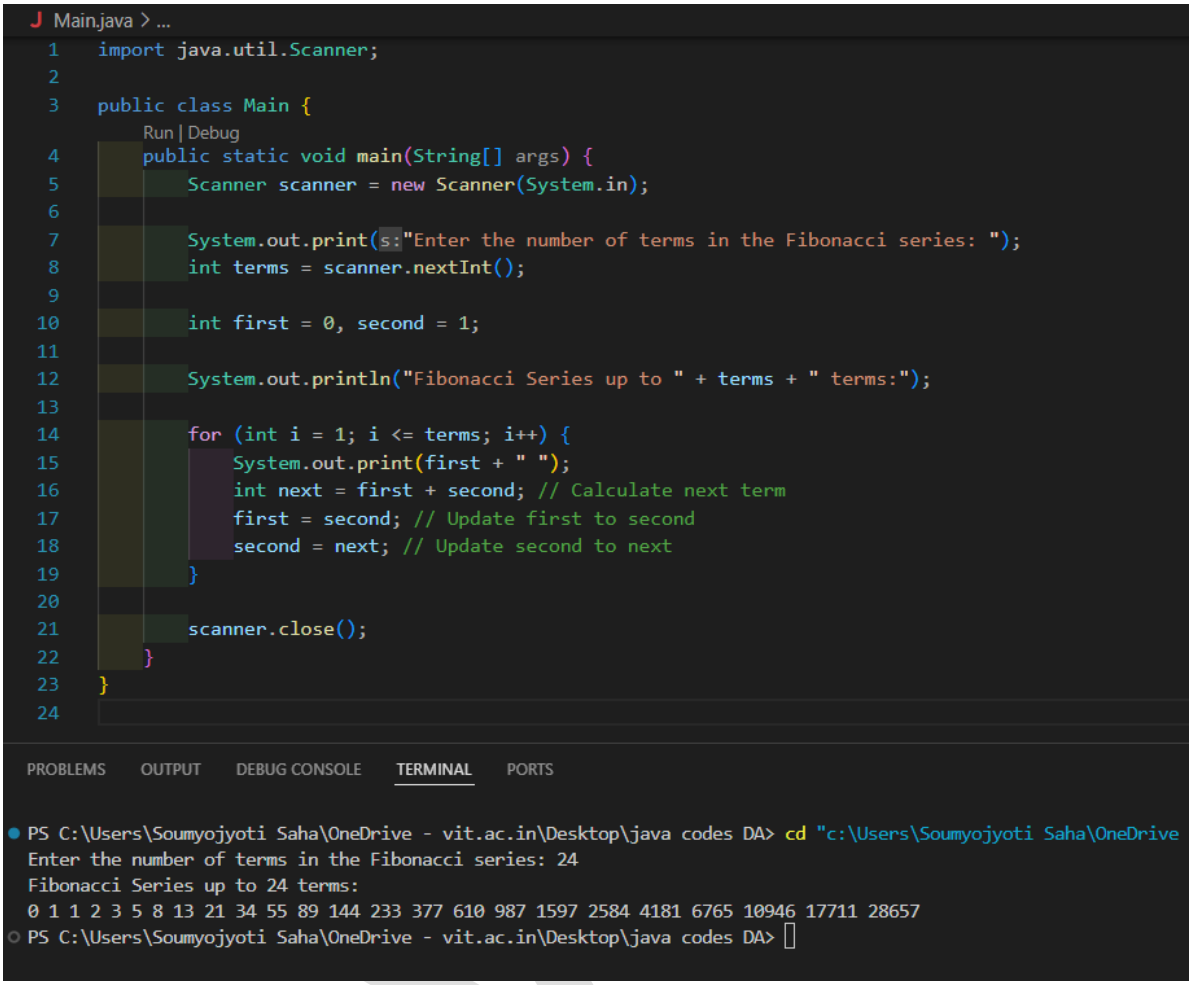
Aim: To write a Java program that prints the Fibonacci series up to a specified number of terms, where each term is the sum of the two preceding ones, starting from 0 and 1.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of terms in the Fibonacci series: ");
        int terms = scanner.nextInt();


        int first = 0, second = 1;


        System.out.println("Fibonacci Series up to " + terms + " terms:");


        for (int i = 1; i <= terms; i++) {
            System.out.print(first + " ");
            int next = first + second; // Calculate next term
            first = second; // Update first to second
            second = next; // Update second to next
        }


        scanner.close();
    }
```

}

Input / Output:



```java
import java.util.Scanner;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Enter the number of terms in the Fibonacci series: ");
        int terms = scanner.nextInt();

        int first = 0, second = 1;

        System.out.println("Fibonacci Series up to " + terms + " terms:");

        for (int i = 1; i <= terms; i++) {
            System.out.print(first + " ");
            int next = first + second; // Calculate next term
            first = second; // Update first to second
            second = next; // Update second to next
        }

        scanner.close();
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the number of terms in the Fibonacci series: 24
Fibonacci Series up to 24 terms:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

9. Write a Java program to take input of different data types from the user and print the inputs.

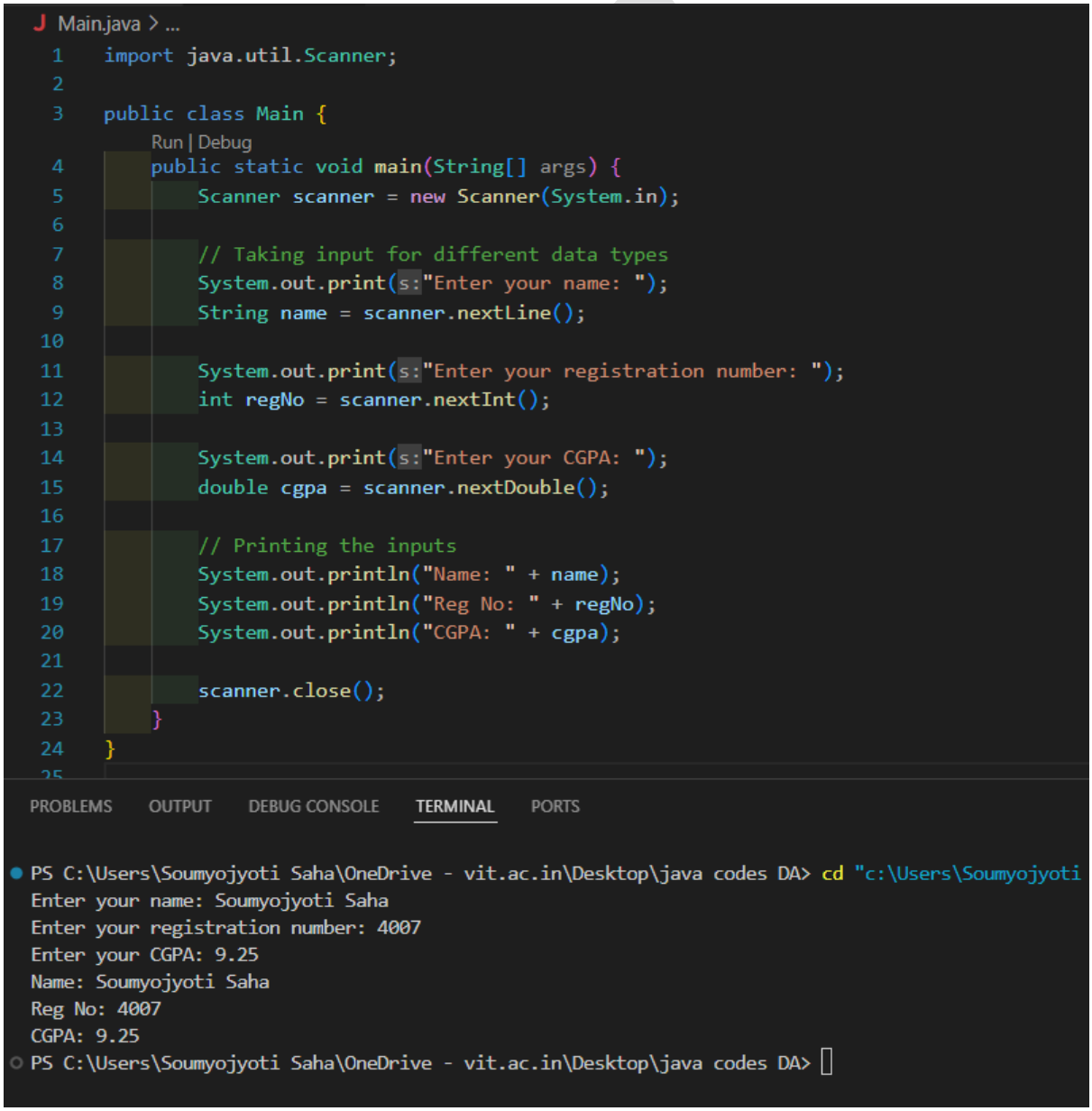
Aim: To write a Java program that takes input of different data types (String, integer, and double) from the user and prints the collected inputs in a formatted manner.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Taking input for different data types
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();


        System.out.print("Enter your registration number: ");
        int regNo = scanner.nextInt();


        System.out.print("Enter your CGPA: ");
        double cgpa = scanner.nextDouble();


        // Printing the inputs
        System.out.println("Name: " + name);
        System.out.println("Reg No: " + regNo);
        System.out.println("CGPA: " + cgpa);
```

```
        scanner.close();
    }
}
```

Input / Output:



10. Write a Java program to get the input from the user for n integers.

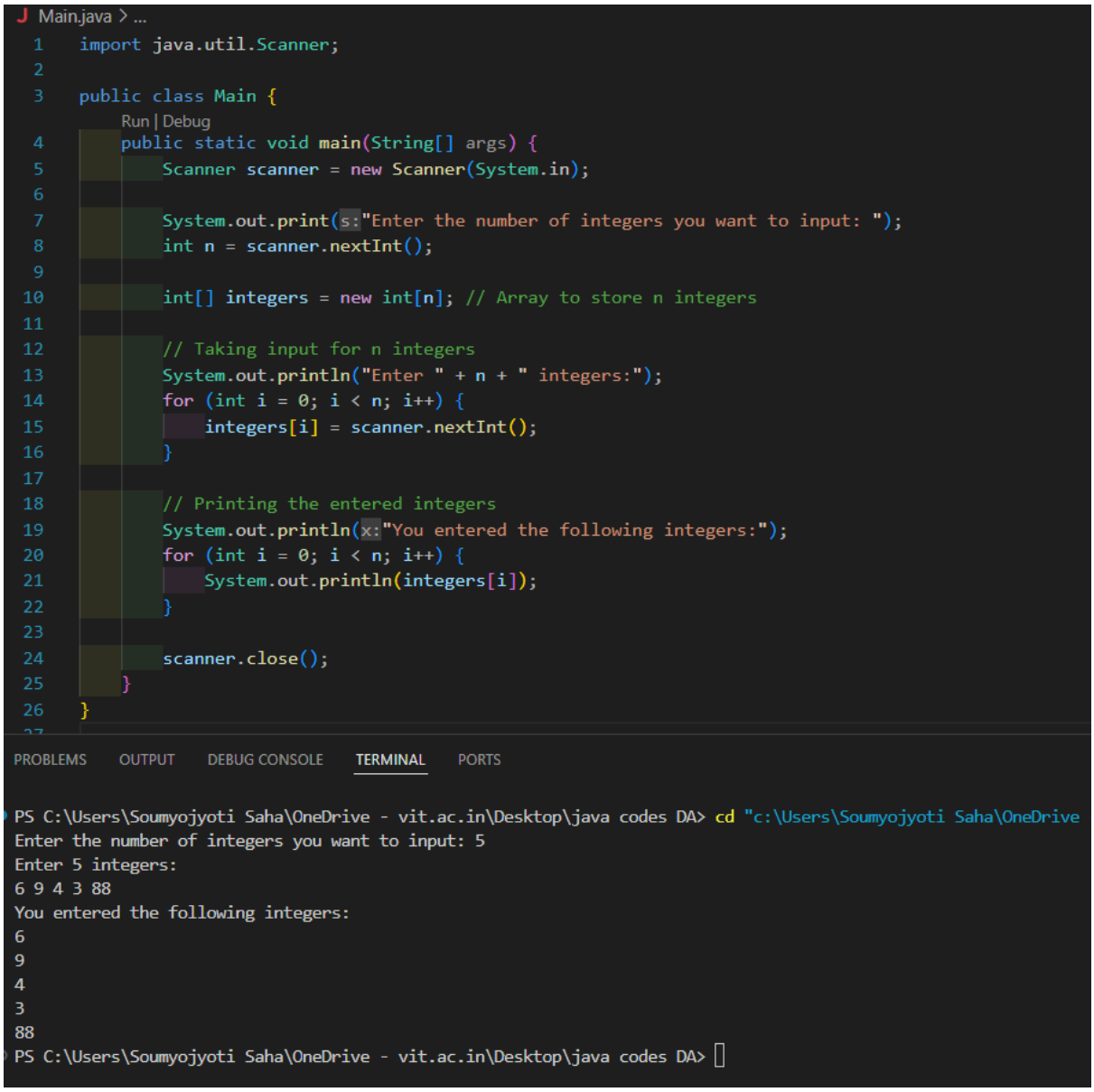
Aim: To write a Java program that prompts the user to enter n integers, stores them in an array, and then prints the entered integers.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of integers you want to input: ");
        int n = scanner.nextInt();


        int[] integers = new int[n]; // Array to store n integers


        // Taking input for n integers
        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            integers[i] = scanner.nextInt();
        }


        // Printing the entered integers
        System.out.println("You entered the following integers:");
        for (int i = 0; i < n; i++) {
            System.out.println(integers[i]);
        }
```

```
        scanner.close();
    }
}
```

Input / Output:



```
Main.java > ...
1   import java.util.Scanner;
2
3   public class Main {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6
7           System.out.print(s:"Enter the number of integers you want to input: ");
8           int n = scanner.nextInt();
9
10          int[] integers = new int[n]; // Array to store n integers
11
12          // Taking input for n integers
13          System.out.println("Enter " + n + " integers:");
14          for (int i = 0; i < n; i++) {
15              integers[i] = scanner.nextInt();
16          }
17
18          // Printing the entered integers
19          System.out.println(x:"You entered the following integers:");
20          for (int i = 0; i < n; i++) {
21              System.out.println(integers[i]);
22          }
23
24          scanner.close();
25      }
26  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the number of integers you want to input: 5
Enter 5 integers:
6 9 4 3 88
You entered the following integers:
6
9
4
3
88
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

11. Write a Java program to print the odd numbers from the array. Get input from the user.

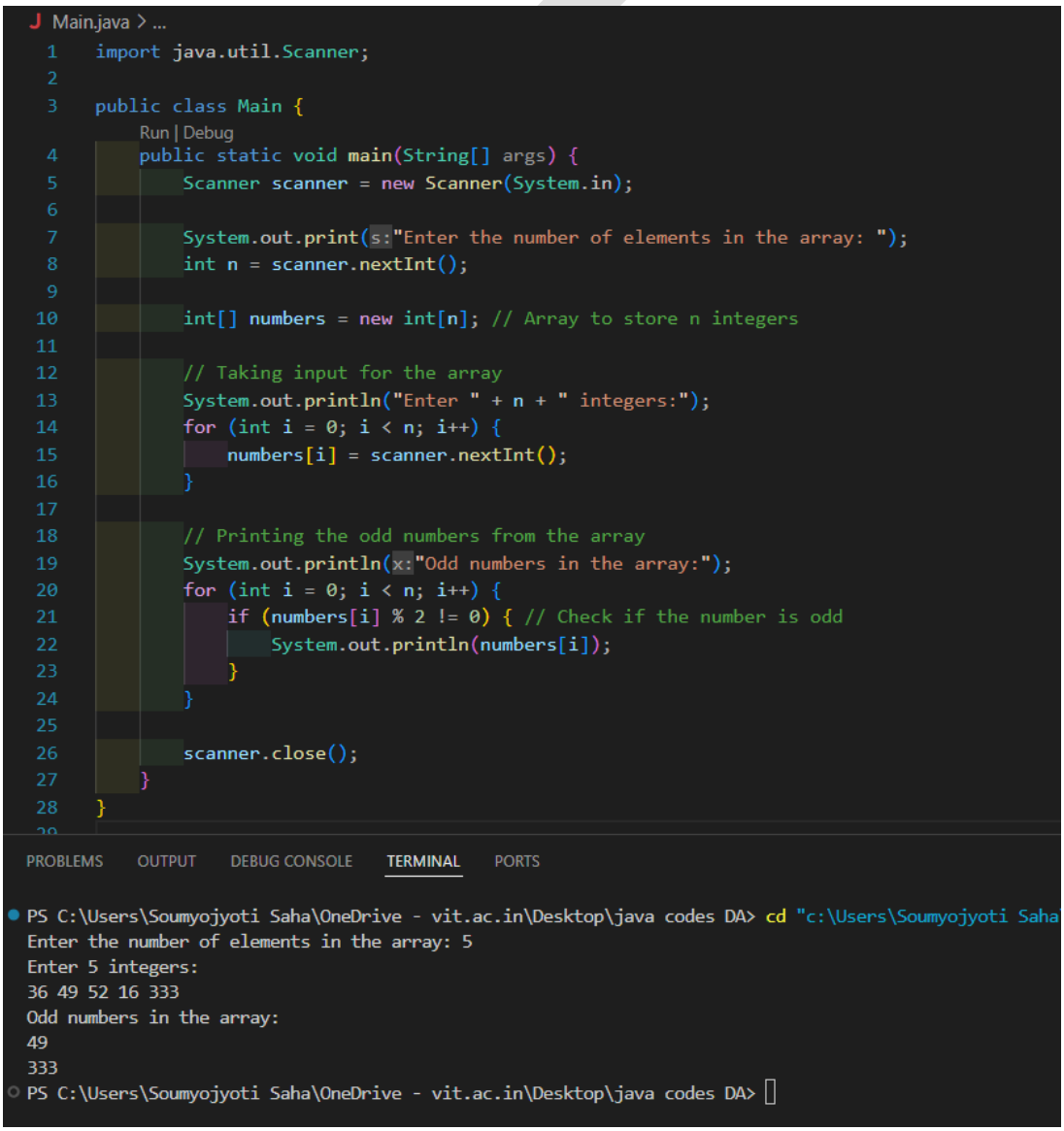
Aim: To write a Java program that takes an array of integers as input from the user and prints the odd numbers contained in that array.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();


        int[] numbers = new int[n]; // Array to store n integers


        // Taking input for the array
        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
        }


        // Printing the odd numbers from the array
        System.out.println("Odd numbers in the array:");
        for (int i = 0; i < n; i++) {
            if (numbers[i] % 2 != 0) { // Check if the number is odd
                System.out.println(numbers[i]);
```

```
        }
    }

    scanner.close();
  }
}
```

Input / Output:



12. Write a Java program to print 2D array. Accept the number of rows and columns and get the inputs from the user.

Aim: To write a Java program that accepts the number of rows and columns for a 2D array, takes inputs for the array elements from the user, and prints the complete 2D array.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Accepting number of rows and columns
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();


        System.out.print("Enter the number of columns: ");
        int cols = scanner.nextInt();


        int[][] array = new int[rows][cols]; // Creating the 2D array


        // Getting input for the 2D array
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print("Element at [" + i + "][" + j + "]: ");
```

```
            array[i][j] = scanner.nextInt();
        }
    }

    // Printing the 2D array
    System.out.println("The 2D array is:");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            System.out.print(array[i][j] + " ");
        }
        System.out.println(); // New line after each row
    }

    scanner.close();
  }
}
```

Input / Output:

Main.java > ...

```java
import java.util.Scanner;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting number of rows and columns
        System.out.print(s:"Enter the number of rows: ");
        int rows = scanner.nextInt();

        System.out.print(s:"Enter the number of columns: ");
        int cols = scanner.nextInt();

        int[][] array = new int[rows][cols]; // Creating the 2D array

        // Getting input for the 2D array
        System.out.println(x:"Enter the elements of the array:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print("Element at [" + i + "][" + j + "]: ");
                array[i][j] = scanner.nextInt();
            }
        }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti
Enter the number of rows: 2
Enter the number of columns: 3
Enter the elements of the array:
Element at [0][0]: 2
Element at [0][1]: 6
Element at [0][2]: 53
Element at [1][0]: 23
Element at [1][1]: 451
Element at [1][2]: 62
The 2D array is:
2 6 53
23 451 62
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

13. Write a Java program to find duplicate values in an array of string values.

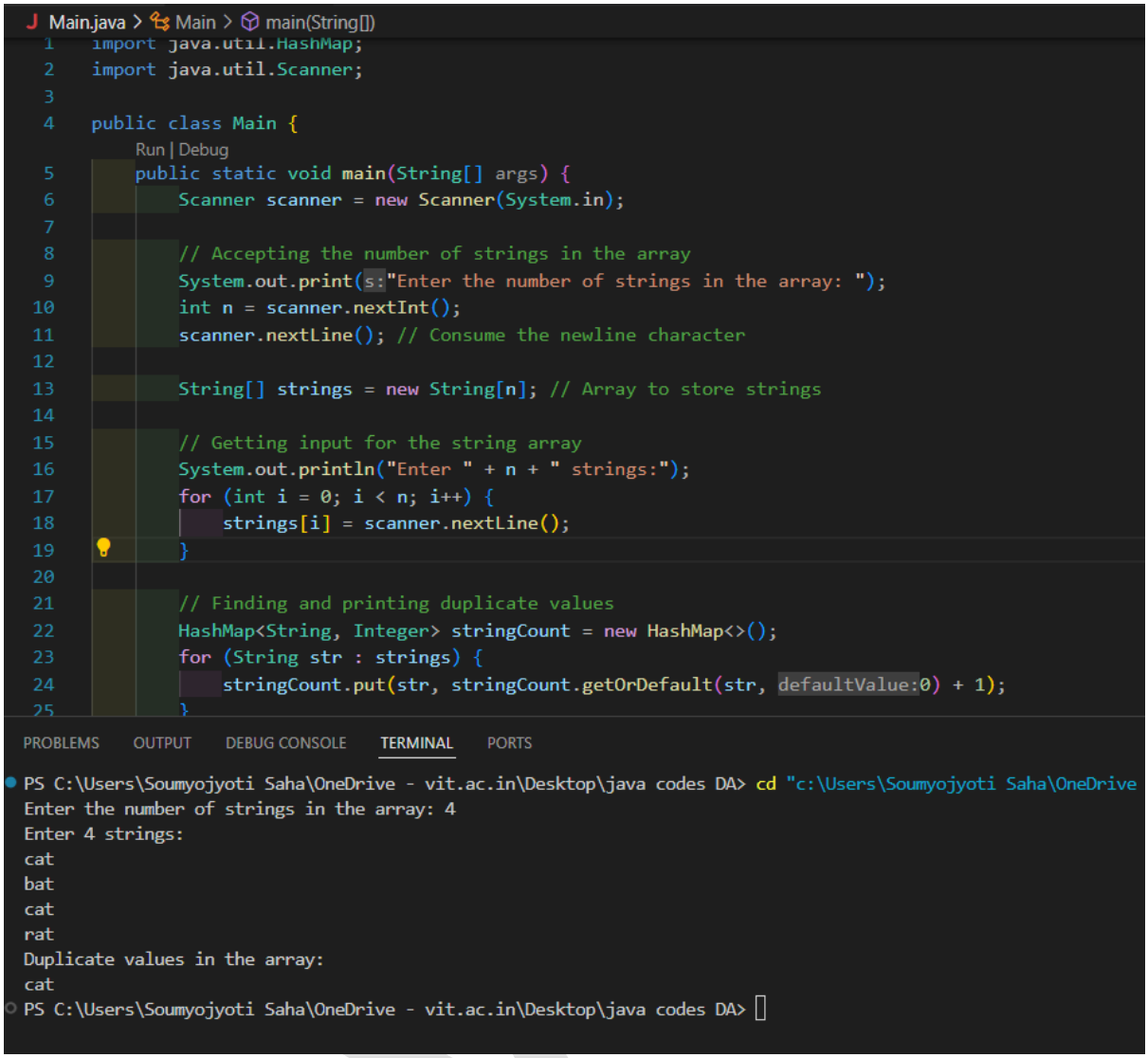
Aim: To write a Java program that accepts an array of string values, identifies any duplicate values, and prints them. If no duplicates are found, it indicates that as well.

Source Code:

```
import java.util.HashMap;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting the number of strings in the array
        System.out.print("Enter the number of strings in the array: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        String[] strings = new String[n]; // Array to store strings

        // Getting input for the string array
        System.out.println("Enter " + n + " strings:");
        for (int i = 0; i < n; i++) {
            strings[i] = scanner.nextLine();
        }

        // Finding and printing duplicate values
```

```
        HashMap<String, Integer> stringCount = new HashMap<>();
        for (String str : strings) {
            stringCount.put(str, stringCount.getOrDefault(str, 0) + 1);
        }

        System.out.println("Duplicate values in the array:");
        boolean hasDuplicates = false;
        for (String key : stringCount.keySet()) {
            if (stringCount.get(key) > 1) {
                System.out.println(key);
                hasDuplicates = true;
            }
        }

        if (!hasDuplicates) {
            System.out.println("No duplicate values found.");
        }

        scanner.close();
    }
}
```

Input / Output:



```
Main.java > Main > main(String[])
import java.util.HashMap;
import java.util.Scanner;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting the number of strings in the array
        System.out.print(s:"Enter the number of strings in the array: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        String[] strings = new String[n]; // Array to store strings

        // Getting input for the string array
        System.out.println("Enter " + n + " strings:");
        for (int i = 0; i < n; i++) {
            strings[i] = scanner.nextLine();
        }

        // Finding and printing duplicate values
        HashMap<String, Integer> stringCount = new HashMap<>();
        for (String str : strings) {
            stringCount.put(str, stringCount.getOrDefault(str, defaultValue:0) + 1);
        }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the number of strings in the array: 4
Enter 4 strings:
cat
bat
cat
rat
Duplicate values in the array:
cat
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

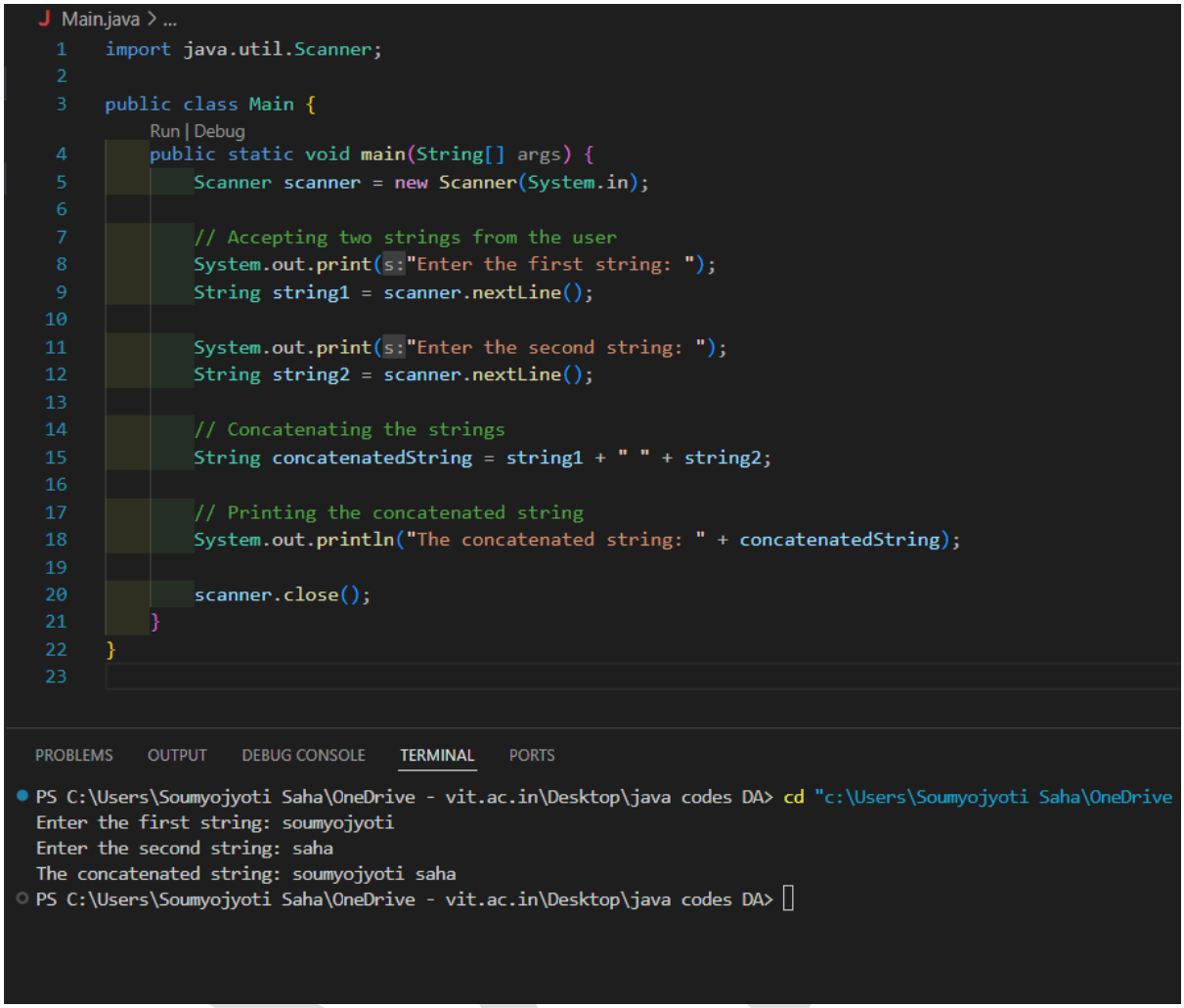
14. Write a Java program to concatenate a given string to the end of another string.

Aim: To write a Java program that accepts two strings from the user, concatenates them, and prints the resulting string. In this example, the output will show the two strings combined with a space in between.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Accepting two strings from the user
        System.out.print("Enter the first string: ");
        String string1 = scanner.nextLine();


        System.out.print("Enter the second string: ");
        String string2 = scanner.nextLine();


        // Concatenating the strings
        String concatenatedString = string1 + " " + string2;


        // Printing the concatenated string
        System.out.println("The concatenated string: " + concatenatedString);


        scanner.close();
    }
}
```

Input / Output:



```
Main.java > ...
1   import java.util.Scanner;
2
3   public class Main {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6
7           // Accepting two strings from the user
8           System.out.print(s:"Enter the first string: ");
9           String string1 = scanner.nextLine();
10
11          System.out.print(s:"Enter the second string: ");
12          String string2 = scanner.nextLine();
13
14          // Concatenating the strings
15          String concatenatedString = string1 + " " + string2;
16
17          // Printing the concatenated string
18          System.out.println("The concatenated string: " + concatenatedString);
19
20          scanner.close();
21      }
22  }
23
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the first string: soumyojyoti
Enter the second string: saha
The concatenated string: soumyojyoti saha
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

15. Write a Java program to compare a given string to another string, ignoring case considerations

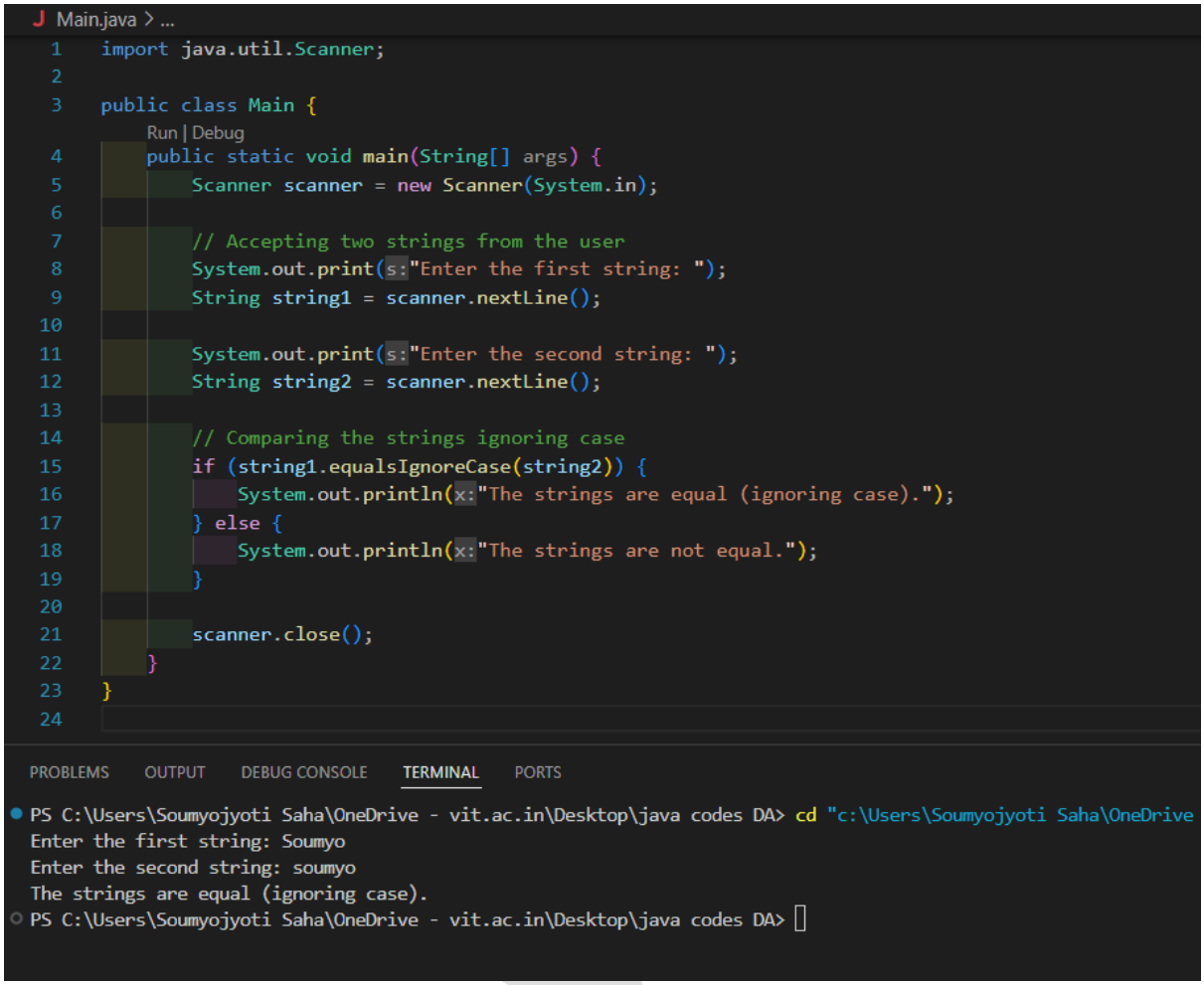
Aim: To write a Java program that compares two user-provided strings, disregarding case sensitivity, and outputs whether the strings are equal or not.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting two strings from the user
        System.out.print("Enter the first string: ");
        String string1 = scanner.nextLine();

        System.out.print("Enter the second string: ");
        String string2 = scanner.nextLine();

        // Comparing the strings ignoring case
        if (string1.equalsIgnoreCase(string2)) {
            System.out.println("The strings are equal (ignoring case).");
        } else {
            System.out.println("The strings are not equal.");
        }

        scanner.close();
```

```
    }
}
```

Input / Output:



```
Main.java > ...
1   import java.util.Scanner;
2
3   public class Main {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6
7           // Accepting two strings from the user
8           System.out.print(s:"Enter the first string: ");
9           String string1 = scanner.nextLine();
10
11          System.out.print(s:"Enter the second string: ");
12          String string2 = scanner.nextLine();
13
14          // Comparing the strings ignoring case
15          if (string1.equalsIgnoreCase(string2)) {
16              System.out.println(x:"The strings are equal (ignoring case).");
17          } else {
18              System.out.println(x:"The strings are not equal.");
19          }
20
21          scanner.close();
22      }
23  }
24

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
Enter the first string: Soumyo
Enter the second string: soumyo
The strings are equal (ignoring case).
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

16. Write a Java program to count the total number of occurrences of a given character in a string without using any loop.

Aim: To write a Java program that counts the total number of occurrences of a given character in a string using recursion, without employing any loops. The program accepts a string and a character from the user and outputs the count of that character in the string.

Source Code:

```
import java.util.Scanner;


public class Main {
    // Recursive method to count occurrences of a character
    public static int countOccurrences(String str, char ch) {
        if (str.isEmpty()) {
            return 0; // Base case: if the string is empty
        } else {
            // Check if the first character matches the given character
            int count = (str.charAt(0) == ch) ? 1 : 0;
            // Recur for the remaining string
            return count + countOccurrences(str.substring(1), ch);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        // Accepting the string and the character to count
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        System.out.print("Enter the character to count: ");
        char characterToCount = scanner.next().charAt(0);

        // Counting occurrences
        int totalOccurrences = countOccurrences(inputString, characterToCount);

        // Printing the result
        System.out.println("Total occurrences of '" + characterToCount + "': " +
totalOccurrences);

        scanner.close();
    }
}
```

Input / Output:

Main.java > Main > main(String[])

```java
import java.util.Scanner;

public class Main {
    // Recursive method to count occurrences of a character
    public static int countOccurrences(String str, char ch) {
        if (str.isEmpty()) {
            return 0; // Base case: if the string is empty
        } else {
            // Check if the first character matches the given character
            int count = (str.charAt(index:0) == ch) ? 1 : 0;
            // Recur for the remaining string
            return count + countOccurrences(str.substring(beginIndex:1), ch);
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting the string and the character to count
        System.out.print(s:"Enter a string: ");
        String inputString = scanner.nextLine();

        System.out.print(s:"Enter the character to count: ");
        char characterToCount = scanner.next().charAt(index:0);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti
Enter a string: Soumyojyoti
Enter the character to count: o
Total occurrences of 'o': 3
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

17. Write a Java program to replace each substring of a given string that matches the given regular expression with the given replacement.

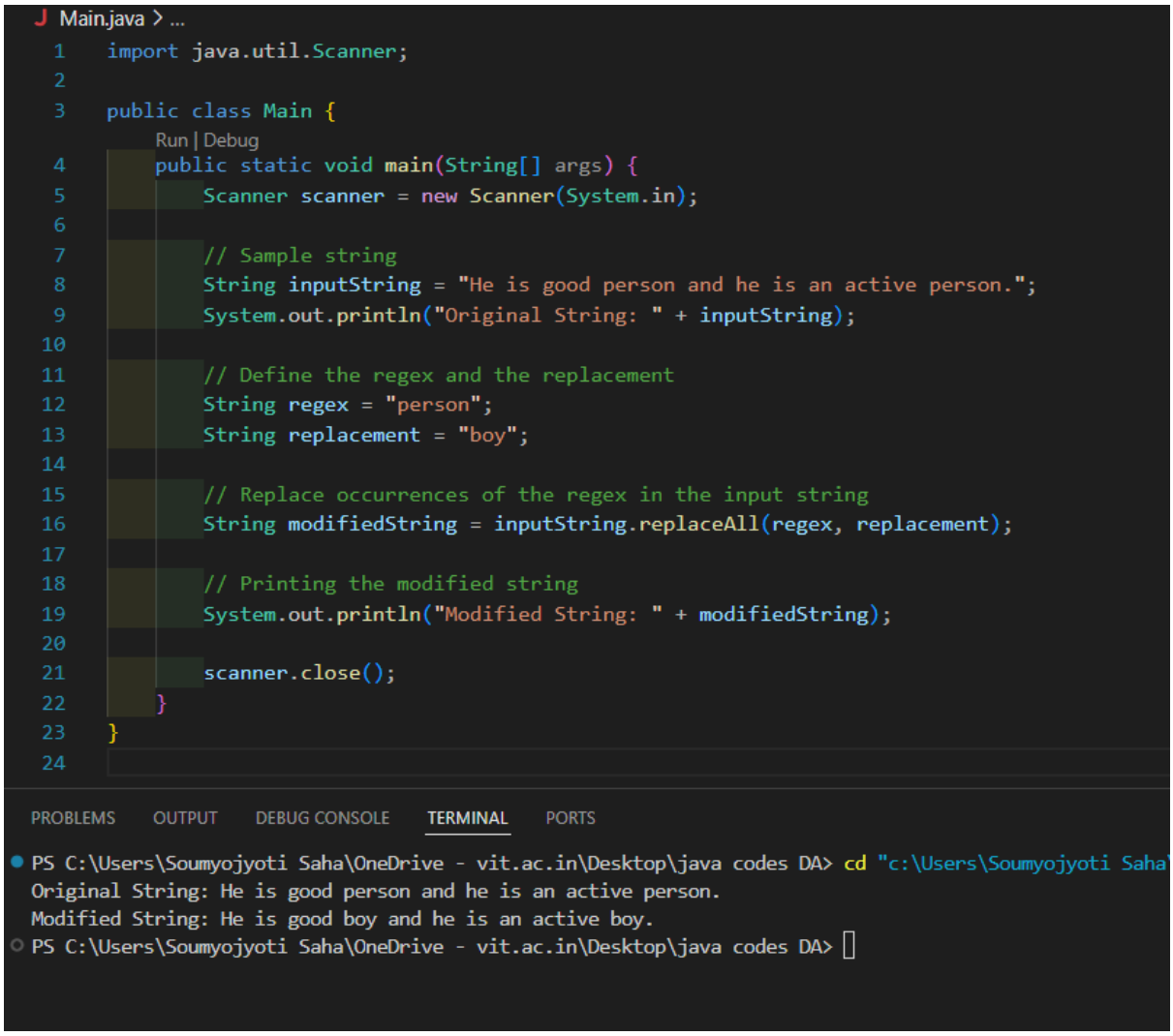
Aim: To write a Java program that replaces all occurrences of substrings in a given string that match a specified regular expression with a designated replacement string. In this example, the program replaces all instances of "person" with "boy" in the provided sample string.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Sample string
        String inputString = "He is good person and he is an active person.";
        System.out.println("Original String: " + inputString);

        // Define the regex and the replacement
        String regex = "person";
        String replacement = "boy";

        // Replace occurrences of the regex in the input string
        String modifiedString = inputString.replaceAll(regex, replacement);

        // Printing the modified string
        System.out.println("Modified String: " + modifiedString);
```

```
        scanner.close();
    }
}
```

Input / Output:



18. To write a Java program that reverses a given string and checks if it is a palindrome, which means it reads the same forwards and backwards. The program outputs the reversed string and indicates whether the original string is a palindrome.

Aim: To write a Java program that reverses a given string and checks if it is a palindrome, which means it reads the same forwards and backwards. The program outputs the reversed string and indicates whether the original string is a palindrome.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting input string from the user
        System.out.print("Enter a string: ");
        String originalString = scanner.nextLine();

        // Reversing the string
        String reversedString = new StringBuilder(originalString).reverse().toString();

        // Checking if the original string is a palindrome
        boolean isPalindrome = originalString.equalsIgnoreCase(reversedString);

        // Printing the results
        System.out.println("Reversed String: " + reversedString);
        if (isPalindrome) {
```

```
            System.out.println(originalString + " is a palindrome.");
        } else {
            System.out.println(originalString + " is not a palindrome.");
        }

        scanner.close();
    }
}
```

Input / Output:

```
Main.java > ...
1   import java.util.Scanner;
2
3   public class Main {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6
7           // Accepting input string from the user
8           System.out.print(s:"Enter a string: ");
9           String originalString = scanner.nextLine();
10
11          // Reversing the string
12          String reversedString = new StringBuilder(originalString).reverse().toString();
13
14          // Checking if the original string is a palindrome
15          boolean isPalindrome = originalString.equalsIgnoreCase(reversedString);
16
17          // Printing the results
18          System.out.println("Reversed String: " + reversedString);
19          if (isPalindrome) {
20              System.out.println(originalString + " is a palindrome.");
21          } else {
22              System.out.println(originalString + " is not a palindrome.");
23          }
24

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\C
Enter a string: Madam
Reversed String: madaM
Madam is a palindrome.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

19. To write a Java program that reverses a given string and checks if it is a palindrome, which means it reads the same forwards and backwards. The program outputs the reversed string and indicates whether the original string is a palindrome.

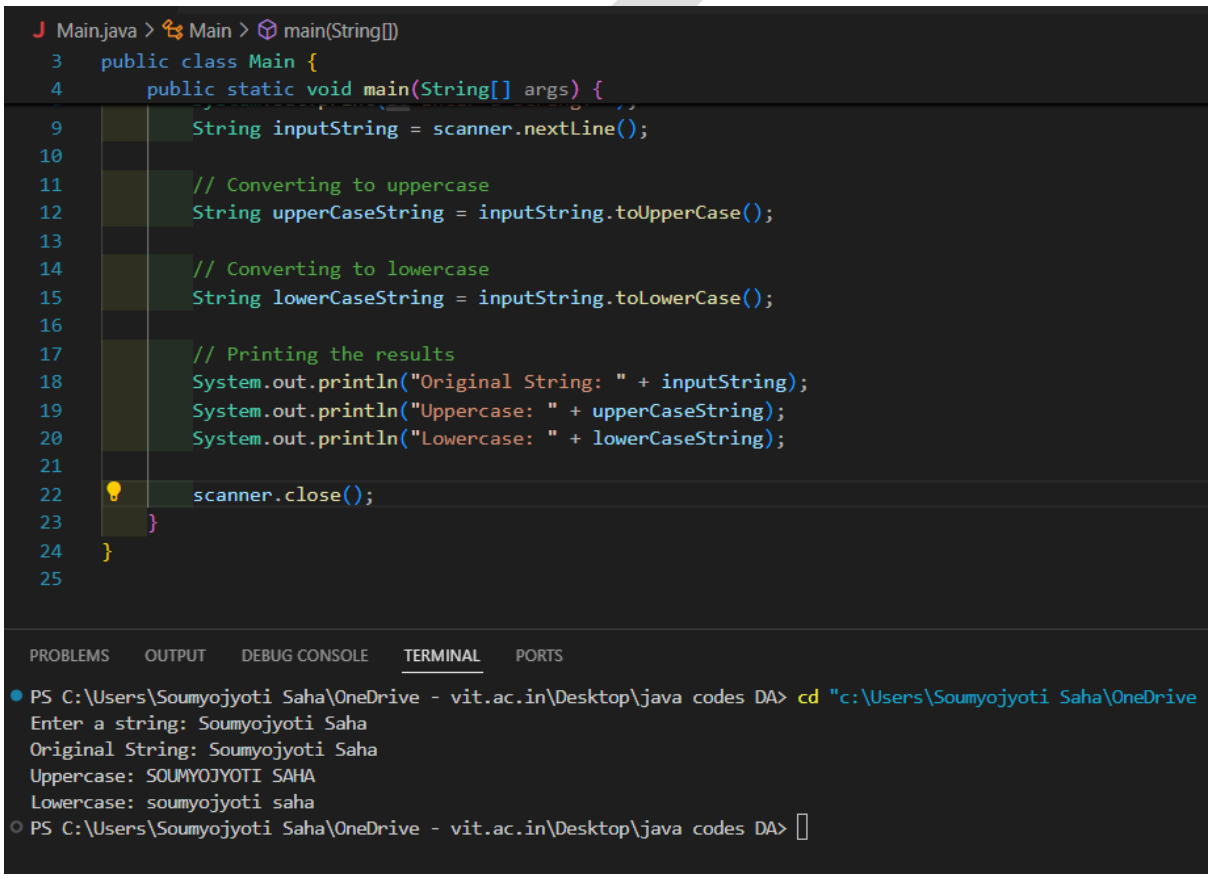
Aim: To write a Java program that takes a user-provided string and converts it to both uppercase and lowercase letters, displaying the original string along with its uppercase and lowercase versions.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting input string from the user
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        // Converting to uppercase
        String upperCaseString = inputString.toUpperCase();

        // Converting to lowercase
        String lowerCaseString = inputString.toLowerCase();

        // Printing the results
        System.out.println("Original String: " + inputString);
        System.out.println("Uppercase: " + upperCaseString);
```

```
        System.out.println("Lowercase: " + lowerCaseString);


        scanner.close();
    }
}
```

Input / Output:



20. Write a Java program to create an array with the columns Student_name, Student_regno and Student_age (minimum of five value entries).

a. Sort and Print the names by length of the name i. Minimum to maximum length ii. Maximum to minimum length

b. Display the student_names that contain "R" at first, or "E" at the end of the string.
c. Display the duplicate student_names with its register number from the list.

d. Display the Students register number whose age is in between 19 and 23.

Aim: To create a structured program that manages student information and performs various operations such as sorting, filtering, and finding duplicates based on specific criteria.

Source Code:

```
import java.util.*;


class Student {
    String name;
    String regNo;
    int age;


    public Student(String name, String regNo, int age) {
        this.name = name;
        this.regNo = regNo;
        this.age = age;
    }
}


public class Main {
    public static void main(String[] args) {
```

```
// Creating an array of students
Student[] students = new Student[5];
students[0] = new Student("Alice", "1001", 20);
students[1] = new Student("Bob", "1002", 22);
students[2] = new Student("Charlie", "1003", 19);
students[3] = new Student("David", "1004", 23);
students[4] = new Student("Edward", "1005", 21);

// a. Sort and print names by length
System.out.println("a. Sorted names by length (Minimum to Maximum):");
Arrays.sort(students, Comparator.comparingInt(s -> s.name.length()));
for (Student student : students) {
    System.out.println(student.name);
}

System.out.println("\na. Sorted names by length (Maximum to Minimum):");
Arrays.sort(students, (s1, s2) -> Integer.compare(s2.name.length(), s1.name.length()));
for (Student student : students) {
    System.out.println(student.name);
}

// b. Display student names that contain "R" at first or "E" at the end
System.out.println("\nb. Student names with 'R' at first or 'E' at the end:");
for (Student student : students) {
    if (student.name.startsWith("R") || student.name.endsWith("E")) {
        System.out.println(student.name);
    }
```

```
        }

        // c. Display duplicate student names with their registration number
        System.out.println("\nc. Duplicate student names with their registration number:");
        Map<String, List<String>> nameMap = new HashMap<>();
        for (Student student : students) {
            nameMap.computeIfAbsent(student.name, k -> new
ArrayList<>()).add(student.regNo);
        }
        boolean hasDuplicates = false;
        for (Map.Entry<String, List<String>> entry : nameMap.entrySet()) {
            if (entry.getValue().size() > 1) {
                System.out.println("Name: " + entry.getKey() + ", Reg No: " + entry.getValue());
                hasDuplicates = true;
            }
        }
        if (!hasDuplicates) {
            System.out.println("No duplicate names found.");
        }

        // d. Display students' registration numbers whose age is between 19 and 23
        System.out.println("\nd. Registration numbers of students whose age is between 19 and
23:");
        for (Student student : students) {
            if (student.age >= 19 && student.age <= 23) {
                System.out.println(student.regNo);
            }
        }
```

```
    }
}
```

Input / Output:

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive -
a. Sorted names by length (Minimum to Maximum):
Bob
Alice
David
Edward
Charlie

a. Sorted names by length (Maximum to Minimum):
Charlie
Edward
Alice
David
Bob

b. Student names with 'R' at first or 'E' at the end:

c. Duplicate student names with their registration number:
No duplicate names found.

d. Registration numbers of students whose age is between 19 and 23:
1003
1005
1001
1004
1002
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

21. Write a Java program to create a menu driven program to display the delivery time and location of different delivery partners where you have to select between Amazon, Flipkart and other delivery options

Aim: To create a menu-driven Java program that allows users to select a delivery partner and view the delivery time and location associated with their choice, providing an interactive experience.

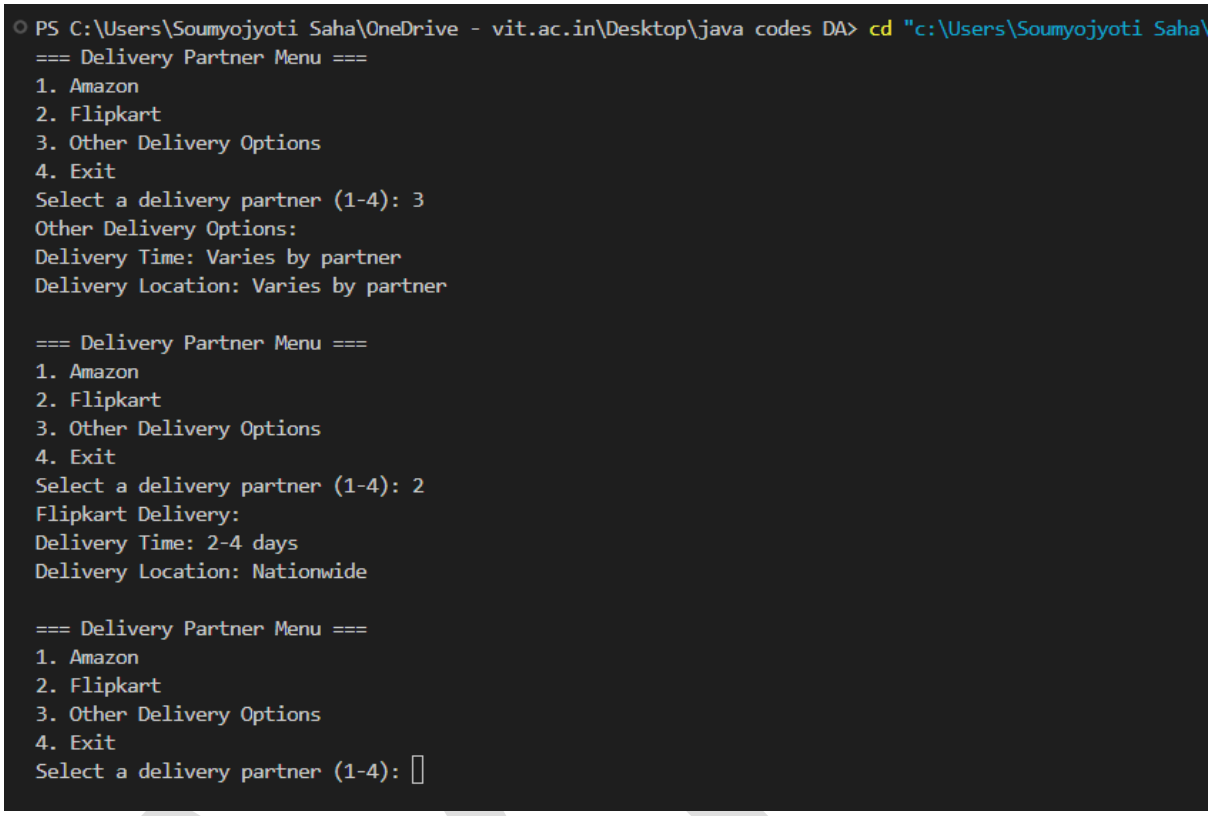
Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            // Display the menu
            System.out.println("=== Delivery Partner Menu ===");
            System.out.println("1. Amazon");
            System.out.println("2. Flipkart");
            System.out.println("3. Other Delivery Options");
            System.out.println("4. Exit");
            System.out.print("Select a delivery partner (1-4): ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            // Display delivery information based on the choice
            switch (choice) {
```

```
            case 1:
                System.out.println("Amazon Delivery:");
                System.out.println("Delivery Time: 1-2 days");
                System.out.println("Delivery Location: Nationwide");
                break;
            case 2:
                System.out.println("Flipkart Delivery:");
                System.out.println("Delivery Time: 2-4 days");
                System.out.println("Delivery Location: Nationwide");
                break;
            case 3:
                System.out.println("Other Delivery Options:");
                System.out.println("Delivery Time: Varies by partner");
                System.out.println("Delivery Location: Varies by partner");
                break;
            case 4:
                System.out.println("Exiting the program. Thank you!");
                break;
            default:
                System.out.println("Invalid choice. Please select between 1-4.");
        }
        System.out.println(); // Blank line for better readability

    } while (choice != 4); // Loop until the user chooses to exit

    scanner.close();
}
```

}

Input / Output:



22. Write a Java program with constructor to find the sum of 'n' integers.

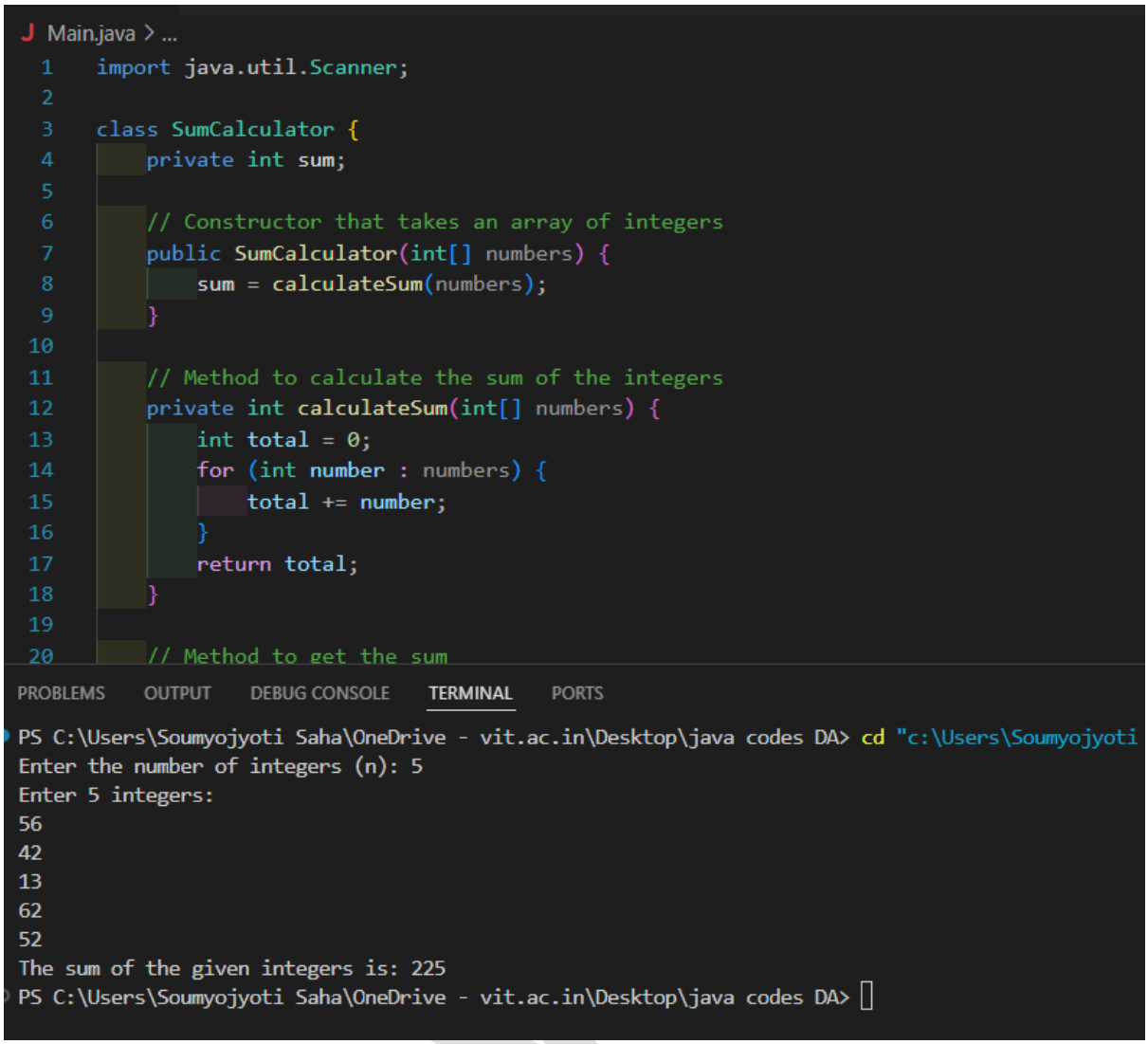
Aim: To demonstrate the use of a constructor in a Java program that calculates the sum of n integers provided by the user.

Source Code:

```
import java.util.Scanner;


class SumCalculator {
    private int sum;

    // Constructor that takes an array of integers
    public SumCalculator(int[] numbers) {
        sum = calculateSum(numbers);
    }

    // Method to calculate the sum of the integers
    private int calculateSum(int[] numbers) {
        int total = 0;
        for (int number : numbers) {
            total += number;
        }
        return total;
    }

    // Method to get the sum
    public int getSum() {
        return sum;
    }
```

```
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting the number of integers
        System.out.print("Enter the number of integers (n): ");
        int n = scanner.nextInt();

        int[] numbers = new int[n];

        // Accepting the integers from the user
        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
        }

        // Creating an instance of SumCalculator
        SumCalculator calculator = new SumCalculator(numbers);

        // Displaying the sum
        System.out.println("The sum of the given integers is: " + calculator.getSum());

        scanner.close();
    }
}
```

Input / Output:



Main.java > ...

```java
import java.util.Scanner;

class SumCalculator {
    private int sum;

    // Constructor that takes an array of integers
    public SumCalculator(int[] numbers) {
        sum = calculateSum(numbers);
    }

    // Method to calculate the sum of the integers
    private int calculateSum(int[] numbers) {
        int total = 0;
        for (int number : numbers) {
            total += number;
        }
        return total;
    }

    // Method to get the sum
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti
Enter the number of integers (n): 5
Enter 5 integers:
56
42
13
62
52
The sum of the given integers is: 225
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

23. Write a Java program with constructor to check whether the given number is even number or odd number.

Aim: Write a Java program with constructor to check whether the given number is even number or odd number.

Source Code:

```
import java.util.Scanner;


class NumberChecker {
    private int number;

    // Constructor that takes an integer
    public NumberChecker(int number) {
        this.number = number;
    }

    // Method to check if the number is even or odd
    public String checkEvenOrOdd() {
        if (number % 2 == 0) {
            return number + " is an even number.";
        } else {
            return number + " is an odd number.";
        }
    }
}

public class Main {
```

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accepting a number from the user
        System.out.print("Enter a number: ");
        int inputNumber = scanner.nextInt();

        // Creating an instance of NumberChecker
        NumberChecker numberChecker = new NumberChecker(inputNumber);

        // Displaying whether the number is even or odd
        System.out.println(numberChecker.checkEvenOrOdd());

        scanner.close();
    }
}
```

Input / Output:

Main.java > ...

```java
import java.util.Scanner;

class NumberChecker {
    private int number;

    // Constructor that takes an integer
    public NumberChecker(int number) {
        this.number = number;
    }

    // Method to check if the number is even or odd
    public String checkEvenOrOdd() {
        if (number % 2 == 0) {
            return number + " is an even number.";
        } else {
            return number + " is an odd number.";
        }
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti
Enter a number: 62
62 is an even number.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti
Enter a number: 35
35 is an odd number.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

24. Write a Java program to create a class A and inherit class B from class A. Both of them have the same method display(). Create a new main class ExampleDemo and call the display method in class B.

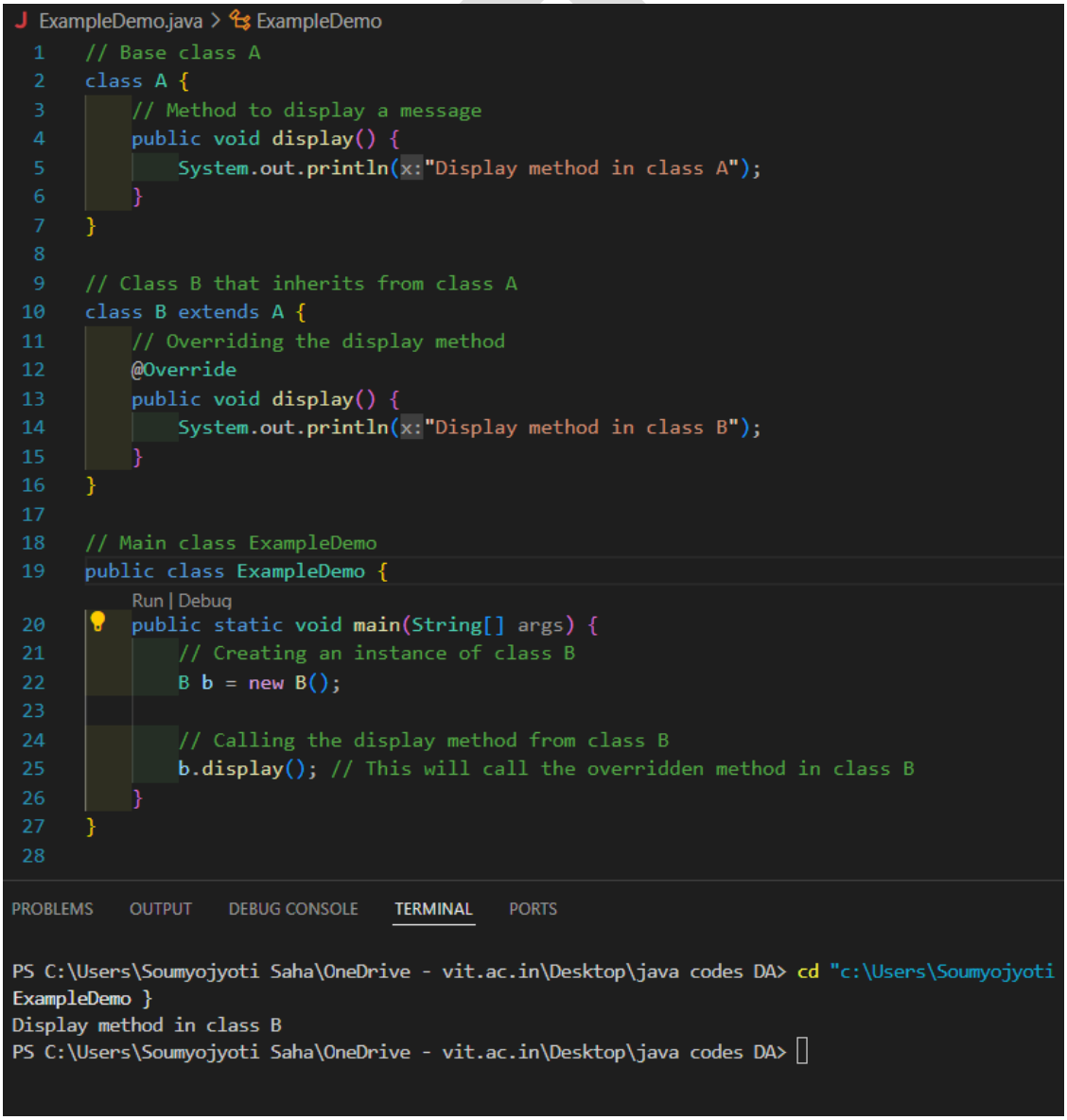
Aim: To demonstrate inheritance in Java by creating a base class with a method and an inherited class that overrides the same method, showing how polymorphism works in action.

Source Code:

```
// Base class A
class A {
    // Method to display a message
    public void display() {
        System.out.println("Display method in class A");
    }
}

// Class B that inherits from class A
class B extends A {
    // Overriding the display method
    @Override
    public void display() {
        System.out.println("Display method in class B");
    }
}

// Main class ExampleDemo
public class ExampleDemo {
    public static void main(String[] args) {
```

```
        // Creating an instance of class B
        B b = new B();

        // Calling the display method from class B
        b.display(); // This will call the overridden method in class B
    }
}
```

Input / Output:



25. Write a menu driven program to (use string functions), a. Appends a string to the end of another string b. Converts a string to lower case letters c. Converts a string to upper case letters d. Returns the length of a specified string

Aim: To create a menu-driven Java program that utilizes string functions for various operations such as appending, case conversion, and length calculation, providing an interactive experience for the user.

Source Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;
        String str1 = "", str2 = "";

        do {
            // Display the menu
            System.out.println("=== String Operations Menu ===");
            System.out.println("1. Append a string to the end of another string");
            System.out.println("2. Convert a string to lower case letters");
            System.out.println("3. Convert a string to upper case letters");
            System.out.println("4. Return the length of a specified string");
            System.out.println("5. Exit");
            System.out.print("Select an option (1-5): ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character
```

```
switch (choice) {
    case 1:
        // Appends a string to the end of another string
        System.out.print("Enter the first string: ");
        str1 = scanner.nextLine();
        System.out.print("Enter the string to append: ");
        str2 = scanner.nextLine();
        str1 += str2; // Append str2 to str1
        System.out.println("Resulting String: " + str1);
        break;

    case 2:
        // Converts a string to lower case letters
        System.out.print("Enter a string: ");
        str1 = scanner.nextLine();
        System.out.println("Lowercase String: " + str1.toLowerCase());
        break;

    case 3:
        // Converts a string to upper case letters
        System.out.print("Enter a string: ");
        str1 = scanner.nextLine();
        System.out.println("Uppercase String: " + str1.toUpperCase());
        break;

    case 4:
        // Returns the length of a specified string
```

```
                System.out.print("Enter a string: ");
                str1 = scanner.nextLine();
                System.out.println("Length of the string: " + str1.length());
                break;

            case 5:
                // Exit the program
                System.out.println("Exiting the program. Thank you!");
                break;

            default:
                System.out.println("Invalid choice. Please select between 1-5.");
        }
        System.out.println(); // Blank line for better readability

    } while (choice != 5); // Loop until the user chooses to exit

    scanner.close();
  }
}
```

Input / Output:

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
=== String Operations Menu ===
1. Append a string to the end of another string
2. Convert a string to lower case letters
3. Convert a string to upper case letters
4. Return the length of a specified string
5. Exit
Select an option (1-5): 4
Enter a string: the quick brown fox jumps over a lazy dog
Length of the string: 41

=== String Operations Menu ===
1. Append a string to the end of another string
2. Convert a string to lower case letters
3. Convert a string to upper case letters
4. Return the length of a specified string
5. Exit
Select an option (1-5): 3
Enter a string: cat
Uppercase String: CAT

=== String Operations Menu ===
1. Append a string to the end of another string
2. Convert a string to lower case letters
3. Convert a string to upper case letters
4. Return the length of a specified string
5. Exit
Select an option (1-5): 2
Enter a string: BAT
Lowercase String: bat

=== String Operations Menu ===
1. Append a string to the end of another string
2. Convert a string to lower case letters
3. Convert a string to upper case letters
4. Return the length of a specified string
5. Exit
Select an option (1-5): 1
Enter the first string: cats love fish
Enter the string to append: curry
Resulting String: cats love fishcurry
```

26. Write a Java program to swap the first and last elements of an array (length minimum of 7) and store the result in another array.

Aim: To create a Java program that swaps the first and last elements of an array with a minimum length of 7, storing the result in another array, and displays both the original and modified arrays.

Source Code:

```
import java.util.Arrays;
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Define the minimum length of the array
        final int MIN_LENGTH = 7;
        System.out.print("Enter the number of elements in the array (minimum " +
MIN_LENGTH + "): ");
        int n = scanner.nextInt();


        // Check if the entered length is valid
        while (n < MIN_LENGTH) {
            System.out.print("Please enter a valid length (minimum " + MIN_LENGTH + "): ");
            n = scanner.nextInt();
        }


        // Create an array and accept elements from the user
        int[] originalArray = new int[n];
```

```
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            originalArray[i] = scanner.nextInt();
        }

        // Create a new array to store the result
        int[] swappedArray = new int[n];

        // Swap the first and last elements
        swappedArray[0] = originalArray[n - 1]; // Last element goes to the first position
        swappedArray[n - 1] = originalArray[0]; // First element goes to the last position

        // Copy the remaining elements
        for (int i = 1; i < n - 1; i++) {
            swappedArray[i] = originalArray[i];
        }

        // Display the result
        System.out.println("Original Array: " + Arrays.toString(originalArray));
        System.out.println("Array after swapping first and last elements: " +
Arrays.toString(swappedArray));

        scanner.close();
    }
}
```

Input / Output:

Main.java > Main > main(String[])

```java
import java.util.Arrays;
import java.util.Scanner;

Run | Debug
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Define the minimum length of the array
        final int MIN_LENGTH = 7;
        System.out.print("Enter the number of elements in the array (minimum " + MIN_LENGTH + "): ");
        int n = scanner.nextInt();

        // Check if the entered length is valid
        while (n < MIN_LENGTH) {
            System.out.print("Please enter a valid length (minimum " + MIN_LENGTH + "): ");
            n = scanner.nextInt();
        }

        // Create an array and accept elements from the user
        int[] originalArray = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            originalArray[i] = scanner.nextInt();
        }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive - vit.ac.
Enter the number of elements in the array (minimum 7): 7
Enter 7 elements:
56 25 84 39 24 33 89
Original Array: [56, 25, 84, 39, 24, 33, 89]
Array after swapping first and last elements: [89, 25, 84, 39, 24, 33, 56]
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

27. Write a Java program to add all the digits of a given positive integer array. a. Write a Java program to multiply corresponding elements of two individual integer arrays.

Sample Output: Array1: [1, 3, -5, 4]

Array2: [1, 4, -5, -2]

Result array: [1, 12, 25, -8]

Aim: To create a Java program that adds all the digits of a given positive integer array and multiplies corresponding elements of two integer arrays, displaying the results clearly for the user.

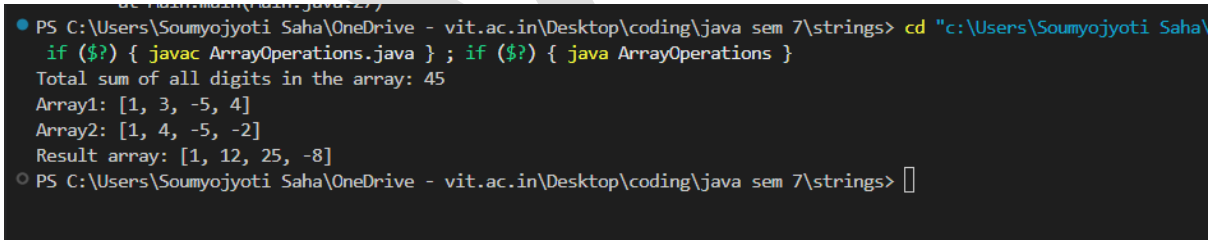
Source Code:

```
public class ArrayOperations {
    public static void main(String[] args) {
        // Aim: To add all digits in a positive integer array and multiply corresponding elements of two integer arrays.

        // Task 1: Add all digits of a positive integer array
        int[] numArray = {123, 456, 789}; // Example array
        int totalDigitSum = addAllDigits(numArray);
        System.out.println("Total sum of all digits in the array: " + totalDigitSum);

        // Task 2: Multiply corresponding elements of two integer arrays
        int[] array1 = {1, 3, -5, 4};
        int[] array2 = {1, 4, -5, -2};
        int[] resultArray = multiplyCorrespondingElements(array1, array2);

        // Display the result
```

```
    System.out.print("Array1: ");
    displayArray(array1);
    System.out.print("Array2: ");
    displayArray(array2);
    System.out.print("Result array: ");
    displayArray(resultArray);
}

// Method to add all digits of a given positive integer array
public static int addAllDigits(int[] array) {
    int sum = 0;
    for (int num : array) {
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }
    }
    return sum;
}

// Method to multiply corresponding elements of two integer arrays
public static int[] multiplyCorrespondingElements(int[] array1, int[] array2) {
    int length = Math.min(array1.length, array2.length);
    int[] resultArray = new int[length];

    for (int i = 0; i < length; i++) {
        resultArray[i] = array1[i] * array2[i];
```

```
        }

        return resultArray;
    }

    // Helper method to display an array
    public static void displayArray(int[] array) {
        System.out.print("[");
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i]);
            if (i < array.length - 1) System.out.print(", ");
        }
        System.out.println("]");
    }
}
```

Input / Output:



28. EXCEPTION HANDLING: 5 PROGRAMS

Program 1: Handling Division by Zero

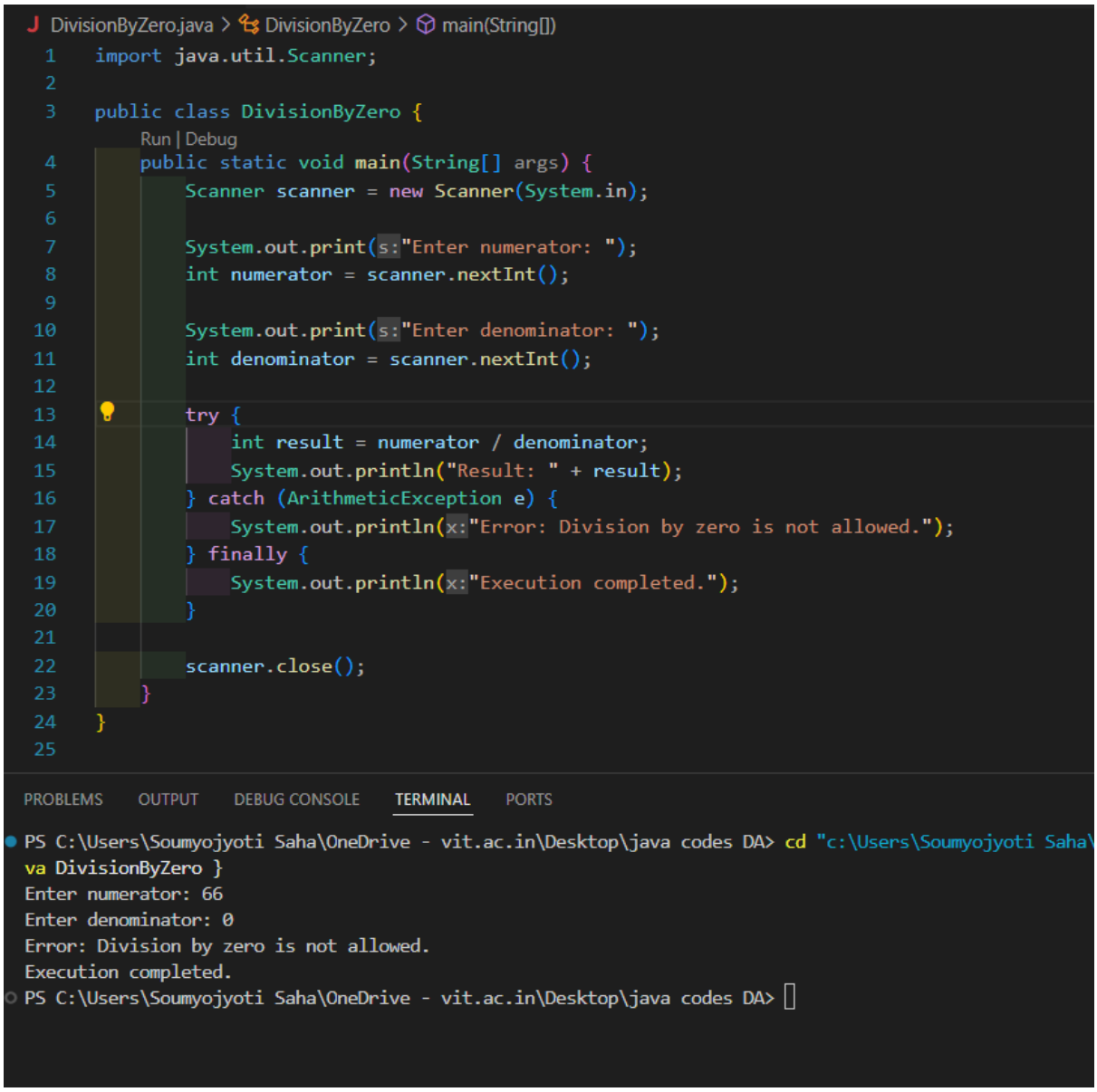
Aim:
To demonstrate the handling of an ArithmeticException caused by division by zero. This program takes two integers as input (numerator and denominator) and performs division while handling the potential exception gracefully, ensuring that the program does not crash and provides a user-friendly error message.

Code:

```
import java.util.Scanner;


public class DivisionByZero {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter numerator: ");
        int numerator = scanner.nextInt();


        System.out.print("Enter denominator: ");
        int denominator = scanner.nextInt();


        try {
            int result = numerator / denominator;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        } finally {
            System.out.println("Execution completed.");
        }
```

```
        scanner.close();
    }
}
```

Input / Output:



Program 2: Handling Array Index Out of Bounds

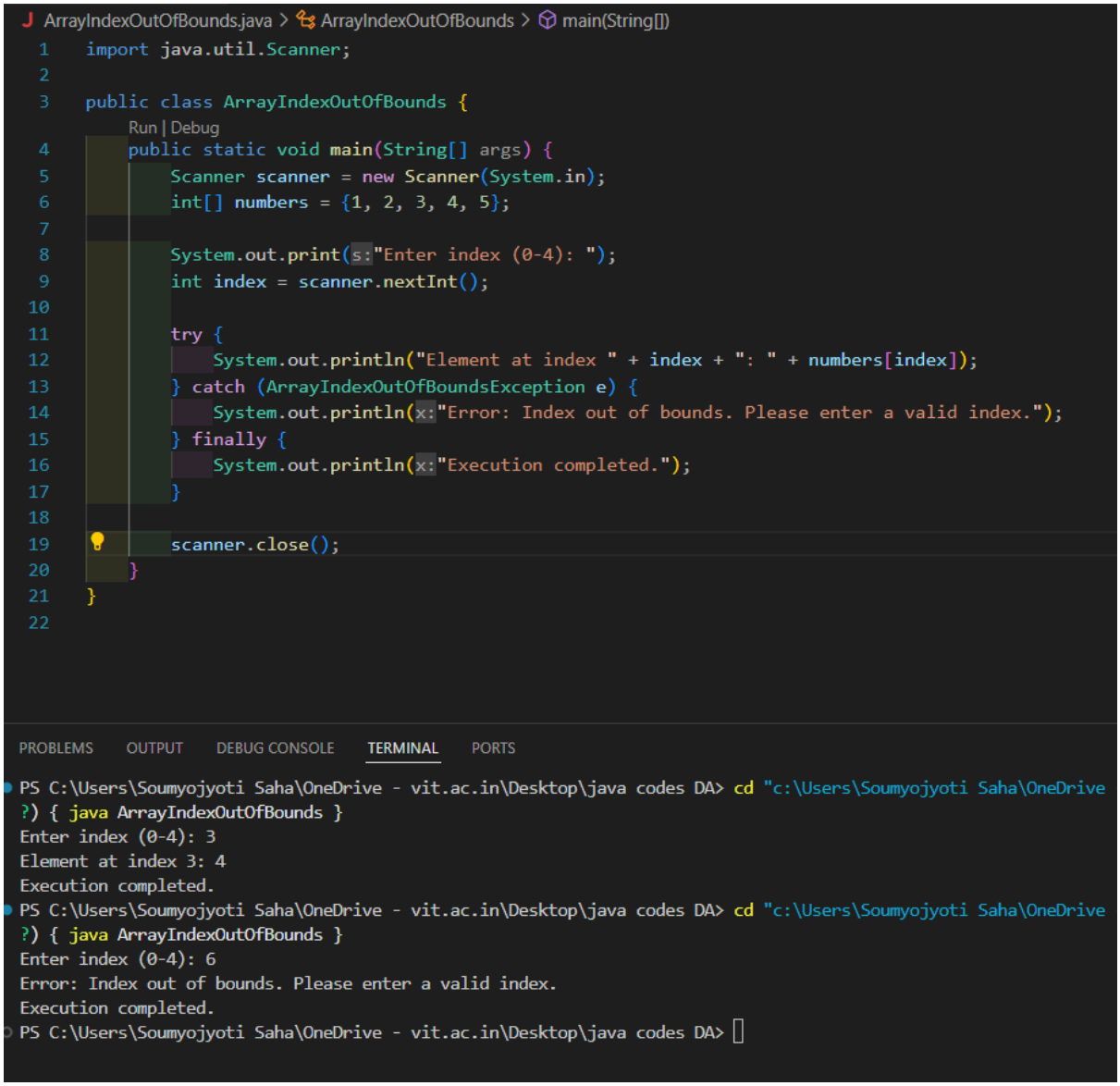
Aim:
To illustrate the handling of an ArrayIndexOutOfBoundsException when accessing elements in an array. The program prompts the user to enter an index and attempts to retrieve the corresponding array element, catching any out-of-bounds access and informing the user of the valid range.

Code:

```
import java.util.Scanner;


public class ArrayIndexOutOfBounds {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = {1, 2, 3, 4, 5};

        System.out.print("Enter index (0-4): ");
        int index = scanner.nextInt();

        try {
            System.out.println("Element at index " + index + ": " + numbers[index]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Index out of bounds. Please enter a valid index.");
        } finally {
            System.out.println("Execution completed.");
        }

        scanner.close();
    }
```

}

Input / Output:



```
ArrayIndexOutOfBounds.java > ArrayIndexOutOfBounds > main(String[])
1   import java.util.Scanner;
2
3   public class ArrayIndexOutOfBounds {
        Run | Debug
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6           int[] numbers = {1, 2, 3, 4, 5};
7
8           System.out.print(s:"Enter index (0-4): ");
9           int index = scanner.nextInt();
10
11          try {
12              System.out.println("Element at index " + index + ": " + numbers[index]);
13          } catch (ArrayIndexOutOfBoundsException e) {
14              System.out.println(x:"Error: Index out of bounds. Please enter a valid index.");
15          } finally {
16              System.out.println(x:"Execution completed.");
17          }
18
19          scanner.close();
20      }
21  }
22

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
?) { java ArrayIndexOutOfBounds }
Enter index (0-4): 3
Element at index 3: 4
Execution completed.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
?) { java ArrayIndexOutOfBounds }
Enter index (0-4): 6
Error: Index out of bounds. Please enter a valid index.
Execution completed.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

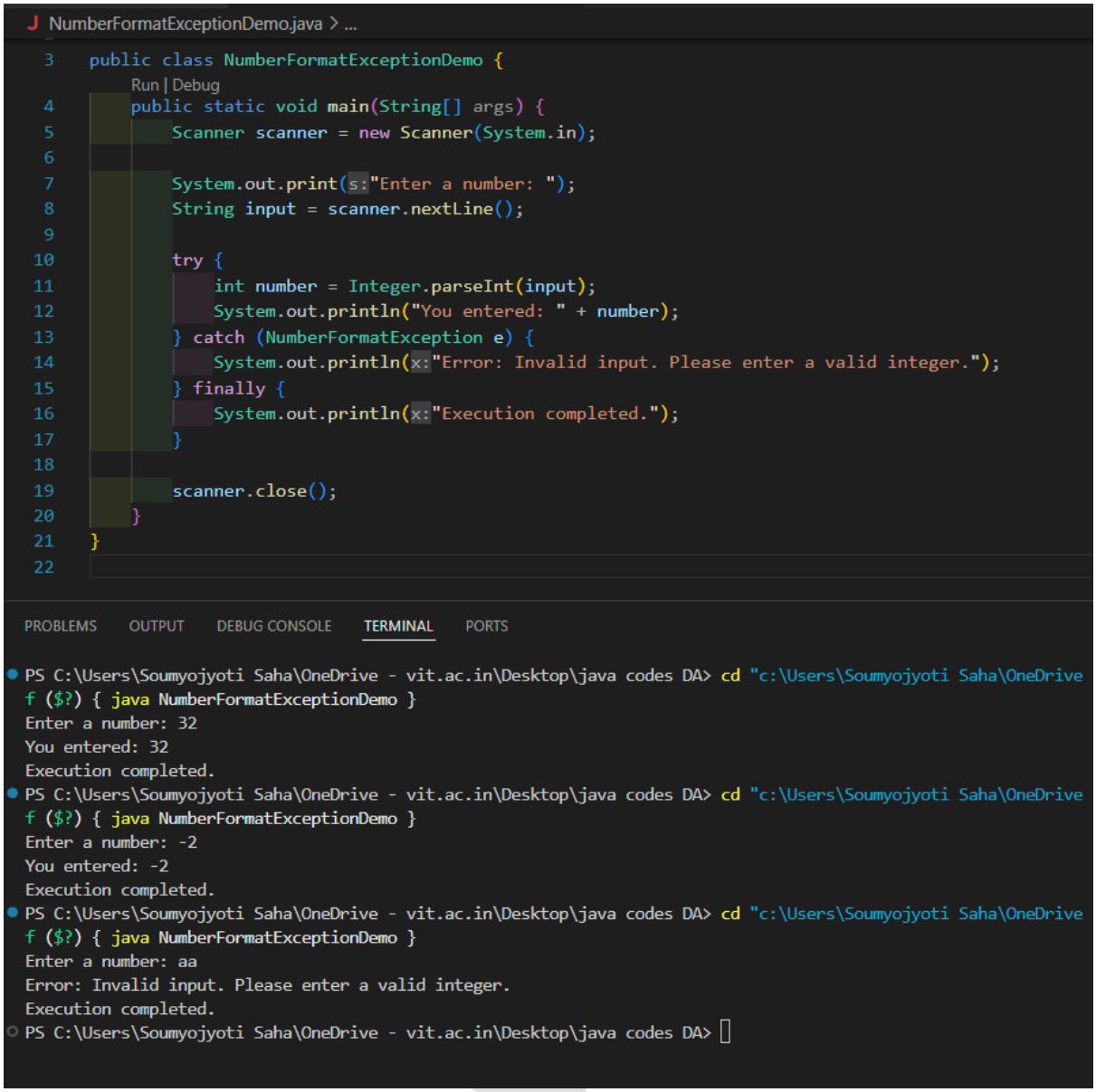
Program 3: Handling Number Format Exception

Aim:
To showcase the handling of a NumberFormatException when parsing a string input to an integer. This program asks the user for a numerical input as a string and attempts to convert it to an integer, catching any format issues and providing feedback on the required input format.

Code:

```
import java.util.Scanner;

public class NumberFormatExceptionDemo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


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

        String input = scanner.nextLine();


        try {

            int number = Integer.parseInt(input);

            System.out.println("You entered: " + number);

        } catch (NumberFormatException e) {

            System.out.println("Error: Invalid input. Please enter a valid integer.");

        } finally {

            System.out.println("Execution completed.");

        }


        scanner.close();

    }

}
```

Input / Output:



```
NumberFormatExceptionDemo.java > ...
```

```java
public class NumberFormatExceptionDemo {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Enter a number: ");
        String input = scanner.nextLine();

        try {
            int number = Integer.parseInt(input);
            System.out.println("You entered: " + number);
        } catch (NumberFormatException e) {
            System.out.println(x:"Error: Invalid input. Please enter a valid integer.");
        } finally {
            System.out.println(x:"Execution completed.");
        }

        scanner.close();
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
f ($?) { java NumberFormatExceptionDemo }
Enter a number: 32
You entered: 32
Execution completed.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
f ($?) { java NumberFormatExceptionDemo }
Enter a number: -2
You entered: -2
Execution completed.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
f ($?) { java NumberFormatExceptionDemo }
Enter a number: aa
Error: Invalid input. Please enter a valid integer.
Execution completed.
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA>
```

Program 4: Handling File Not Found Exception

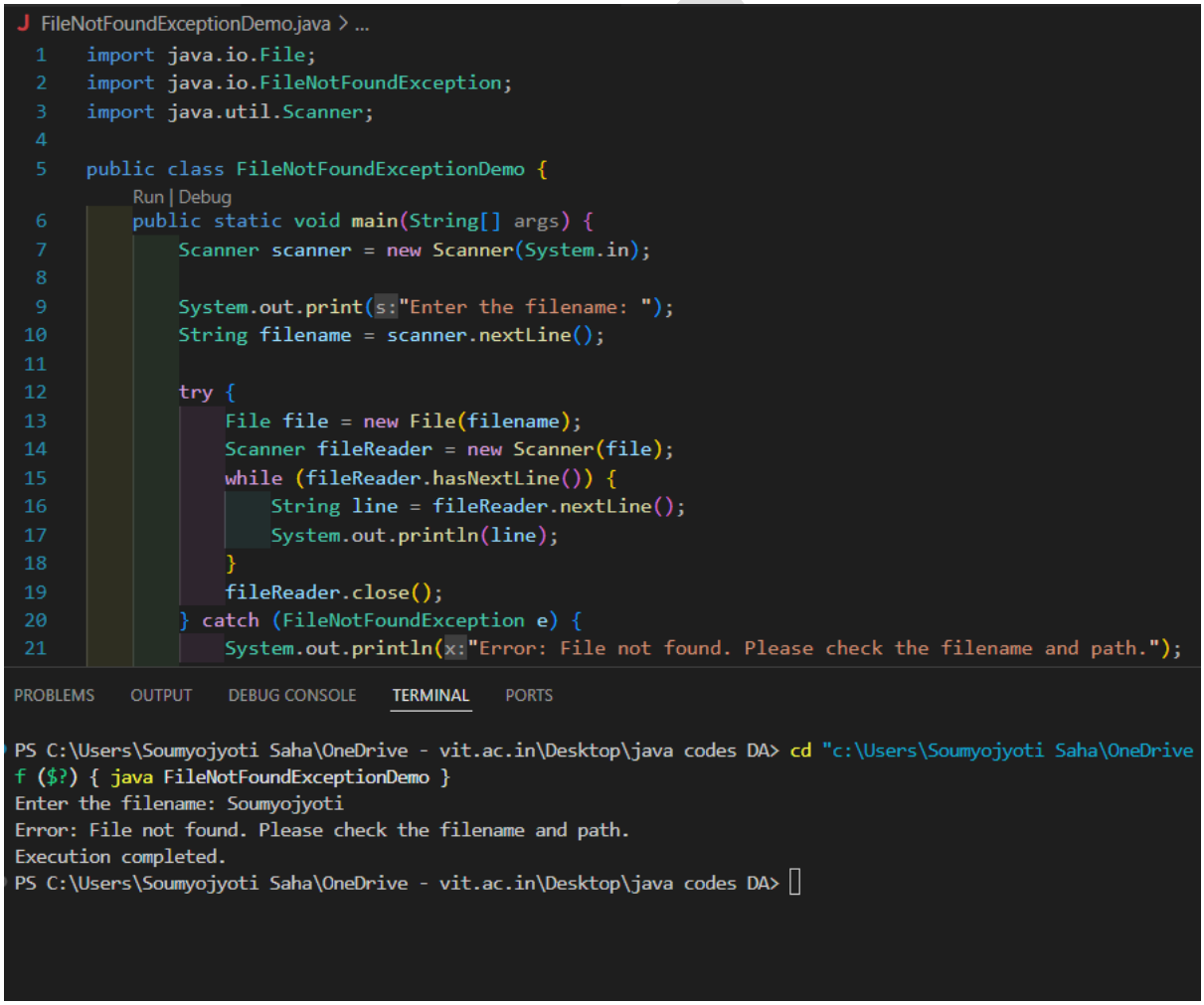
Aim:
To demonstrate the handling of a FileNotFoundException when attempting to read a file. The program prompts the user for a filename and tries to read the file contents, gracefully managing scenarios where the file may not exist and notifying the user accordingly.

Code:

```
import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;


public class FileNotFoundExceptionDemo {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


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

        String filename = scanner.nextLine();


        try {

            File file = new File(filename);

            Scanner fileReader = new Scanner(file);

            while (fileReader.hasNextLine()) {

                String line = fileReader.nextLine();

                System.out.println(line);

            }

            fileReader.close();

        } catch (FileNotFoundException e) {

            System.out.println("Error: File not found. Please check the filename and path.");

        } finally {
```

```
            System.out.println("Execution completed.");
        }


        scanner.close();
    }
}
```

Input / Output:



Program 5: Custom Exception Handling

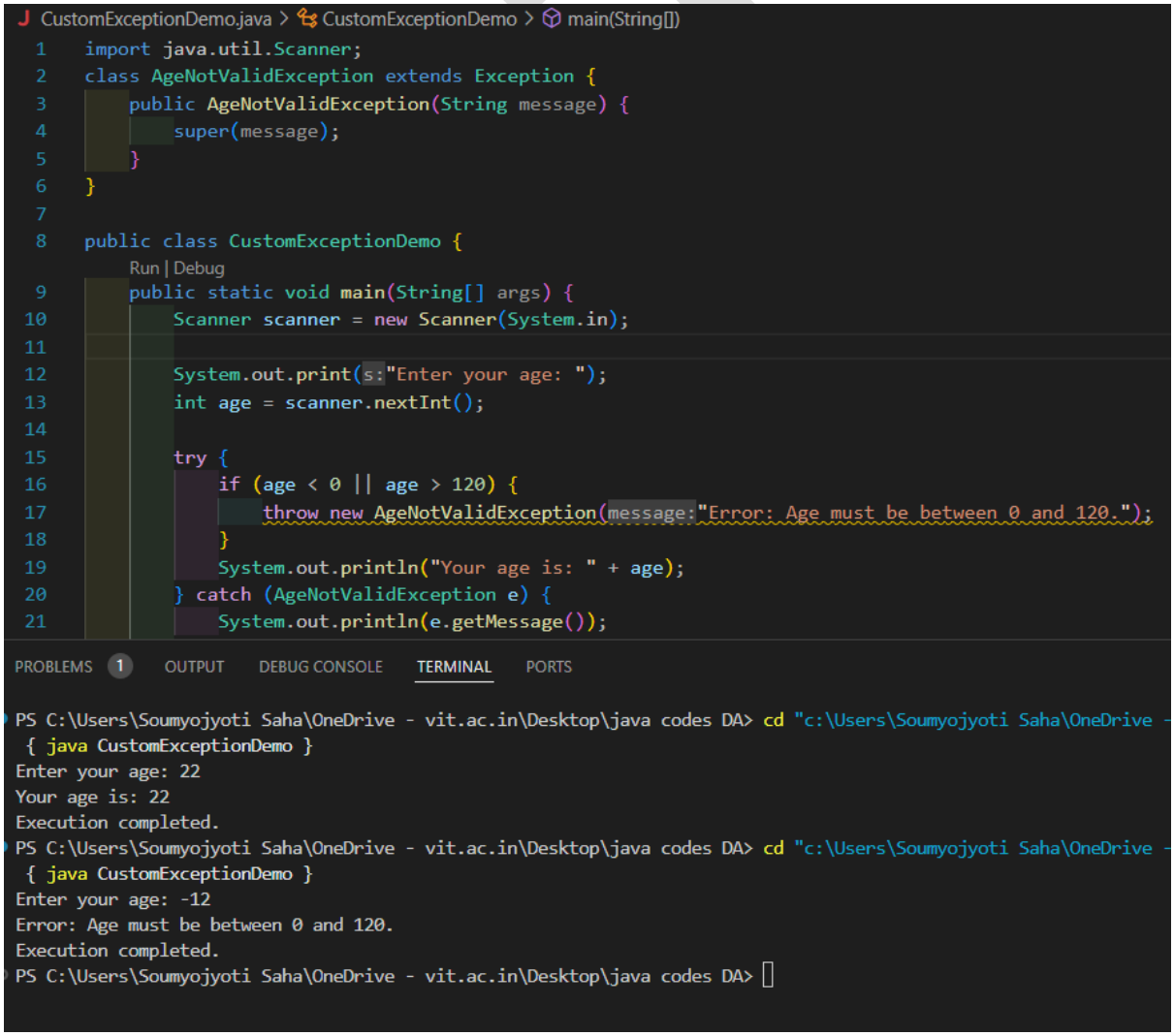
Aim:
To illustrate the creation and handling of a custom exception (AgeNotValidException). The program prompts the user for their age and checks if it falls within a valid range (0 to 120). If the input is invalid, it throws the custom exception and provides a specific error message, demonstrating how to implement user-defined exceptions in Java.

Code:

```
class AgeNotValidException extends Exception {
    public AgeNotValidException(String message) {
        super(message);
    }
}


import java.util.Scanner;


public class CustomExceptionDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter your age: ");
        int age = scanner.nextInt();


        try {
            if (age < 0 || age > 120) {
                throw new AgeNotValidException("Error: Age must be between 0 and 120.");
            }
            System.out.println("Your age is: " + age);
        } catch (AgeNotValidException e) {
```

```
            System.out.println(e.getMessage());
        } finally {
            System.out.println("Execution completed.");
        }

        scanner.close();
    }
}
```

Input / Output:



29. Create a package class A with display function and in a new class outside the package import the package A and print the corresponding output.

Aim: To create a Java package containing a class with a display function and demonstrate how to import and use that class in a different Java file outside the package.

Source Code:

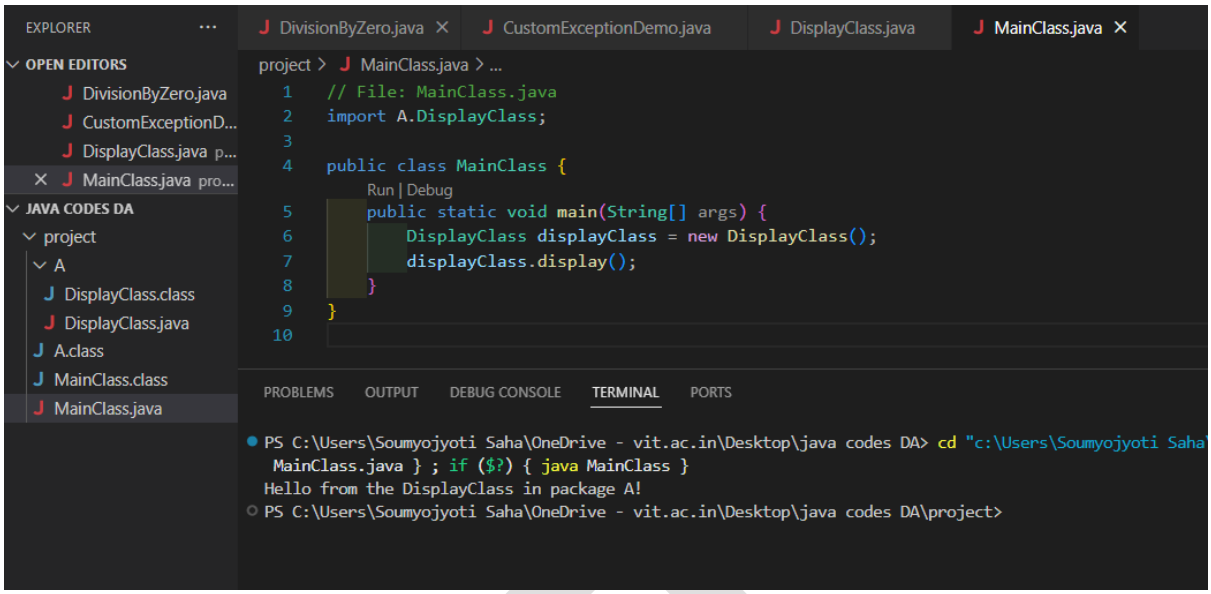
DisplayClass

```
// File: A/DisplayClass.java

package A;


public class DisplayClass {
    public void display() {
        System.out.println("Hello from the DisplayClass in package A!");
    }
}
```

Main Class

```
// File: MainClass.java

import A.DisplayClass;


public class MainClass {
    public static void main(String[] args) {
        DisplayClass displayClass = new DisplayClass();
        displayClass.display();
    }
}
```

Input / Output:



30. Create a package Student with a student name and reg No as attributes, create another package Course having course name and course code as attributes and import both packages in a new java class outside the Student and Course package and display the corresponding outputs.

Aim: To create two Java packages (Student and Course), each containing a class with specific attributes, and demonstrate how to import and use these packages in a separate Java class to display the corresponding outputs.

Source Code:

Package Student: StudentInfo

```
// File: Student/StudentInfo.java

package Student;


public class StudentInfo {
    private String name;
    private String regNo;

    public StudentInfo(String name, String regNo) {
        this.name = name;
        this.regNo = regNo;
    }

    public String getName() {
        return name;
    }

    public String getRegNo() {
        return regNo;
```

```
    }
}
```

Package Course: CourseInfo

```
// File: Course/CourseInfo.java
package Course;

public class CourseInfo {
    private String courseName;
    private String courseCode;

    public CourseInfo(String courseName, String courseCode) {
        this.courseName = courseName;
        this.courseCode = courseCode;
    }

    public String getCourseName() {
        return courseName;
    }

    public String getCourseCode() {
        return courseCode;
    }
}
```

Main Class

```
// File: MainClass.java
import Student.StudentInfo;
```

```
import Course.CourseInfo;


public class MainClass {
    public static void main(String[] args) {
        // Creating a StudentInfo object
        StudentInfo student = new StudentInfo("Rohit", "1001");


        // Creating a CourseInfo object
        CourseInfo course = new CourseInfo("Computer Science", "CS101");


        // Displaying student information
        System.out.println("Student Name: " + student.getName());
        System.out.println("Registration Number: " + student.getRegNo());


        // Displaying course information
        System.out.println("Course Name: " + course.getCourseName());
        System.out.println("Course Code: " + course.getCourseCode());
    }
}
```

Input / Output:

MainClass.java

project > MainClass.java > MainClass

```java
// File: MainClass.java
import Student.StudentInfo;
import Course.CourseInfo;

public class MainClass {
    Run | Debug
    public static void main(String[] args) {
        // Creating a StudentInfo object
        StudentInfo student = new StudentInfo(name:"Rohit", regNo:"1001");

        // Creating a CourseInfo object
        CourseInfo course = new CourseInfo(courseName:"Computer Science", courseCode:"CS101");

        // Displaying student information
        System.out.println("Student Name: " + student.getName());
        System.out.println("Registration Number: " + student.getRegNo());

        // Displaying course information
        System.out.println("Course Name: " + course.getCourseName());
        System.out.println("Course Code: " + course.getCourseCode());
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA> cd "c:\Users\Soumyojyoti Saha\OneDrive
 java MainClass }
Student Name: Rohit
Registration Number: 1001
Course Name: Computer Science
Course Code: CS101
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java codes DA\project>
```

END