

Lab Task 4

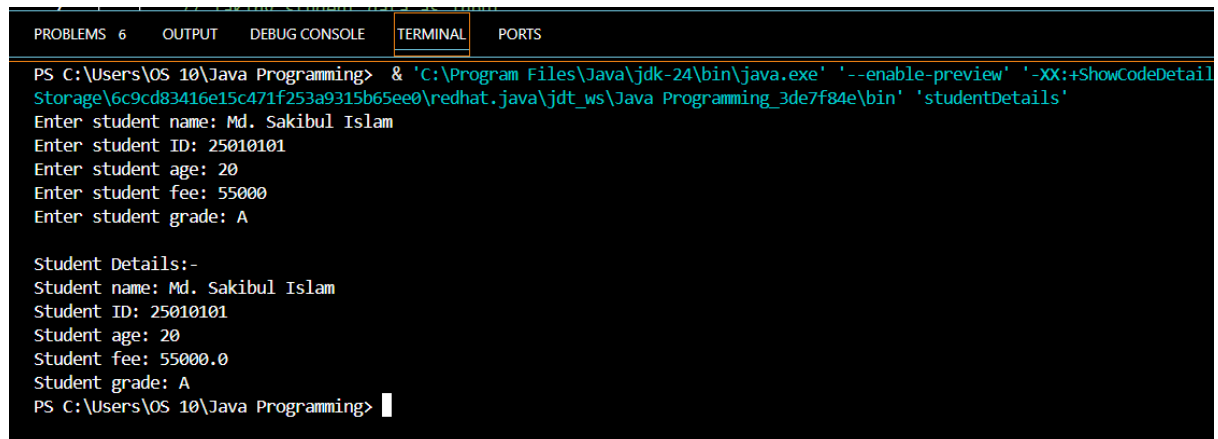
1. Student Details

Input:



```
1 import java.util.Scanner;
2
3 public class studentDetails {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Taking student data as input
8         System.out.print(s:"Enter student name: ");
9         String studentName = scanner.nextLine();
10
11         System.out.print(s:"Enter student ID: ");
12         int studentID = scanner.nextInt();
13
14         System.out.print(s:"Enter student age: ");
15         int studentAge = scanner.nextInt();
16
17         System.out.print(s:"Enter student fee: ");
18         float studentFee = scanner.nextFloat();
19
20         System.out.print(s:"Enter student grade: ");
21         char studentGrade = scanner.next().charAt(index:0);
22
23         // Printing student details
24         System.out.println(x:"\nStudent Details:");
25         System.out.println("Student name: " + studentName);
26         System.out.println("Student ID: " + studentID);
27         System.out.println("Student age: " + studentAge);
28         System.out.println("Student fee: " + studentFee);
29         System.out.println("Student grade: " + studentGrade);
30
31         scanner.close();
32     }
33 }
34
```

Output:



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetails' '-XX:MaxHeapSize=1G' '-Djava.class.path=Storage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\jdt_ws\Java Programming_3de7f84e\bin' 'studentDetails'
Enter student name: Md. Sakibul Islam
Enter student ID: 25010101
Enter student age: 20
Enter student fee: 55000
Enter student grade: A

Student Details:-
Student name: Md. Sakibul Islam
Student ID: 25010101
Student age: 20
Student fee: 55000.0
Student grade: A
PS C:\Users\OS 10\Java Programming>
```

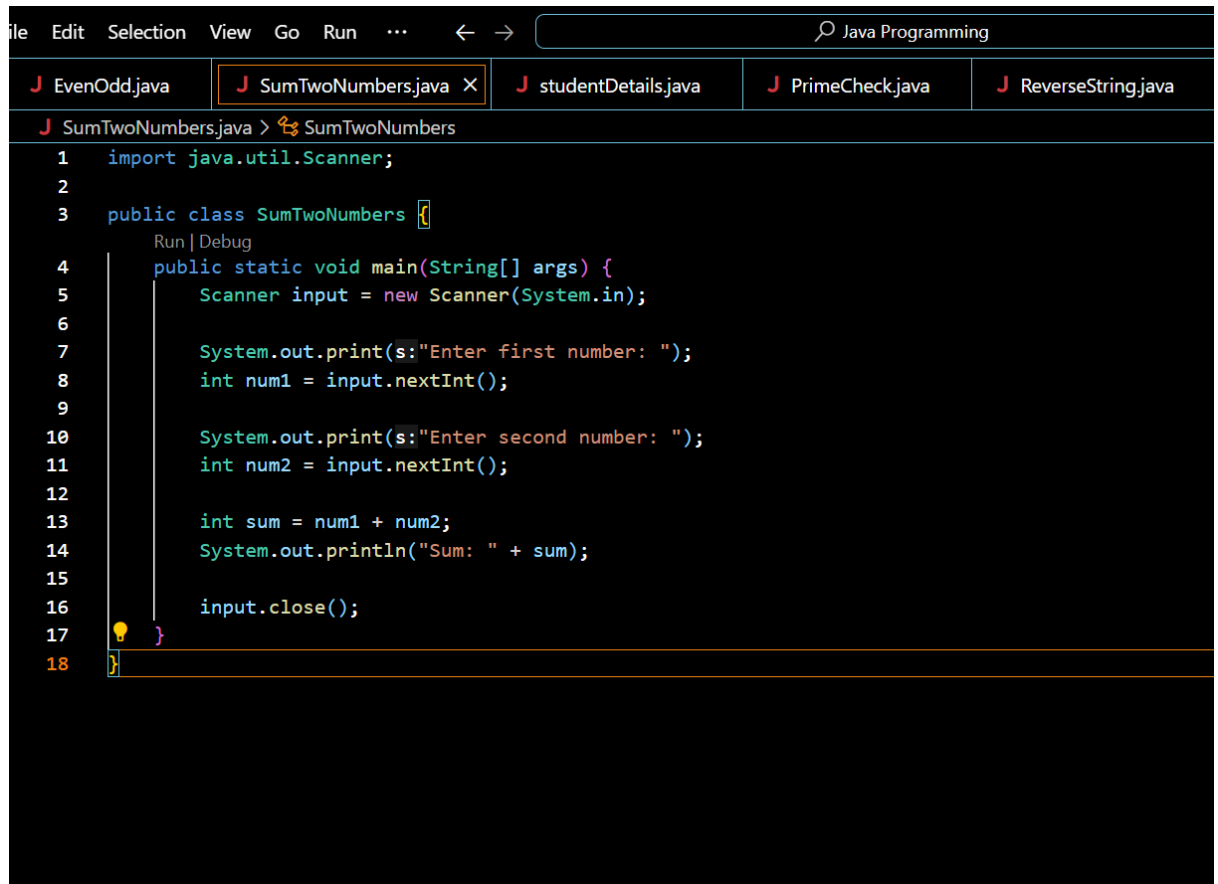
Explanation :

3. Explanation of Important Lines:

- **Scanner scanner = new Scanner(System.in);**
This creates a Scanner object to read input from the console.
- **String studentName = scanner.nextLine();**
Reads the full line entered by the user for the student's name.
- **int studentID = scanner.nextInt();** and other similar lines
Read integer and float inputs respectively for ID, age and fee.
- **char studentGrade = scanner.next().charAt(0);**
Reads a single character (the grade) from the user input. The `.charAt(0)` means take the first character of the entered string.
- **System.out.println(...)** statements
Display the collected student details in an organized manner.
- **scanner.close();**
Closes the Scanner object to free up resources.

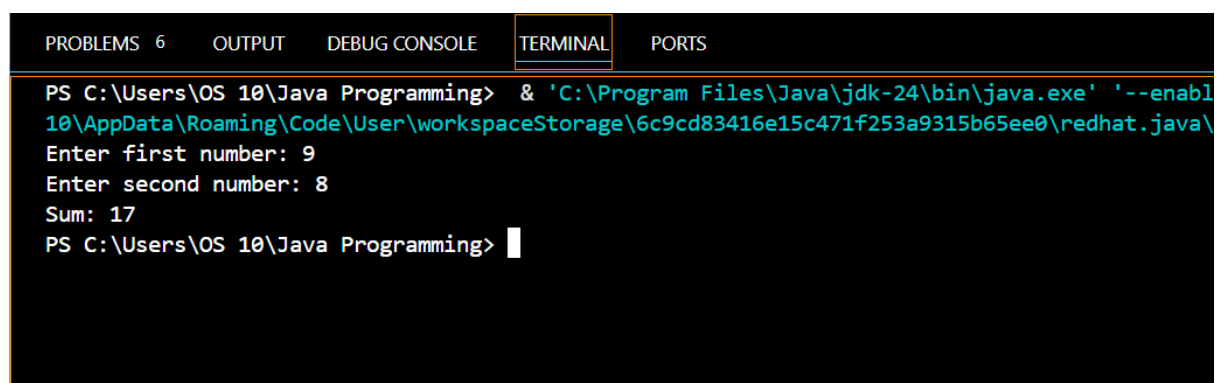
2.Sum of Two Numbers

Input:



```
1 import java.util.Scanner;
2
3 public class SumTwoNumbers {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print(s:"Enter first number: ");
8         int num1 = input.nextInt();
9
10        System.out.print(s:"Enter second number: ");
11        int num2 = input.nextInt();
12
13        int sum = num1 + num2;
14        System.out.println("Sum: " + sum);
15
16        input.close();
17    }
18 }
```

Output:



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enabl
10\AppData\Roaming\Code\User\workspaceStorage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\
Enter first number: 9
Enter second number: 8
Sum: 17
PS C:\Users\OS 10\Java Programming> 
```

Explanation :

Explanation of Important Lines:

- `int number = input.nextInt();`

Reads the integer input from the user.

- `if (number % 2 == 0)`

Checks if the number is divisible by 2 with no remainder (meaning it is even).

- `System.out.println(number + " is Even.");` Prints that the number is even if the condition is true.

- `else { ... }`

If the number is not even, it is odd, and this block handles printing that message.

- `input.close();`

Closes the scanner to free up resources.

4. Find the Factorial of a Number

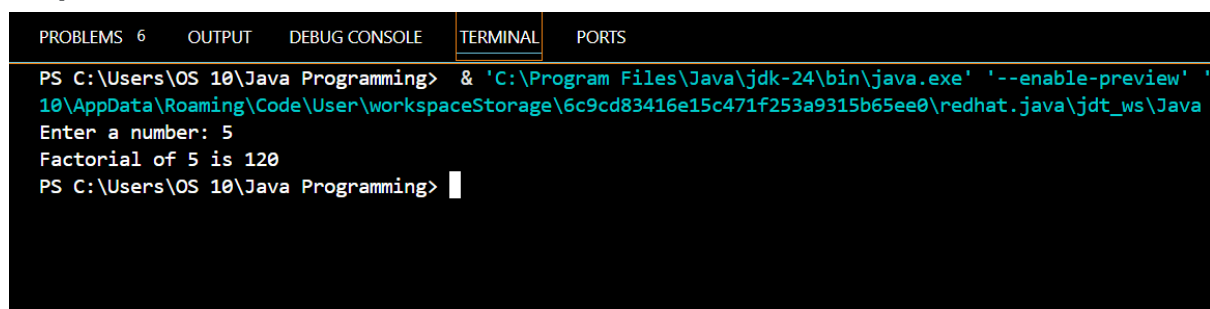
Input:



The screenshot shows an IDE window titled "Java Programming" with several tabs open: EvenOdd.java, SumTwoNumbers.java, studentDetails.java, PrimeCheck.java, ReverseString.java, FibonacciSeries.java, and Factorial.java. The Factorial.java tab is active, displaying the following code:

```
1 import java.util.Scanner;
2
3 public class Factorial {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print(s:"Enter a number: ");
8         int num = input.nextInt();
9         long fact = 1;
10
11         for (int i = 1; i <= num; i++) {
12             fact *= i;
13         }
14
15         System.out.println("Factorial of " + num + " is " + fact);
16         input.close();
17     }
18 }
19
```

Output:



The screenshot shows the IDE's terminal window with the following output:

```
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '
10\AppData\Roaming\Code\User\workspaceStorage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\jdt_ws\Java
Enter a number: 5
Factorial of 5 is 120
PS C:\Users\OS 10\Java Programming>
```

Explanation :

Explanation of Important Lines:

- `int num = input.nextInt();` Reads the integer input from the user.

- `long factorial = 1;`

Initializes the factorial result as 1. We use long here to allow larger factorial values without overflow.

- `for (int i = 1; i <= num; i++) { factorial *= i; }`

This loop runs from 1 to num, multiplying the factorial variable by each i in the range. This effectively calculates the factorial (e.g., $5! = 1 \times 2 \times 3 \times 4 \times 5$).

- `System.out.println("Factorial of " + num + " = " + factorial);`

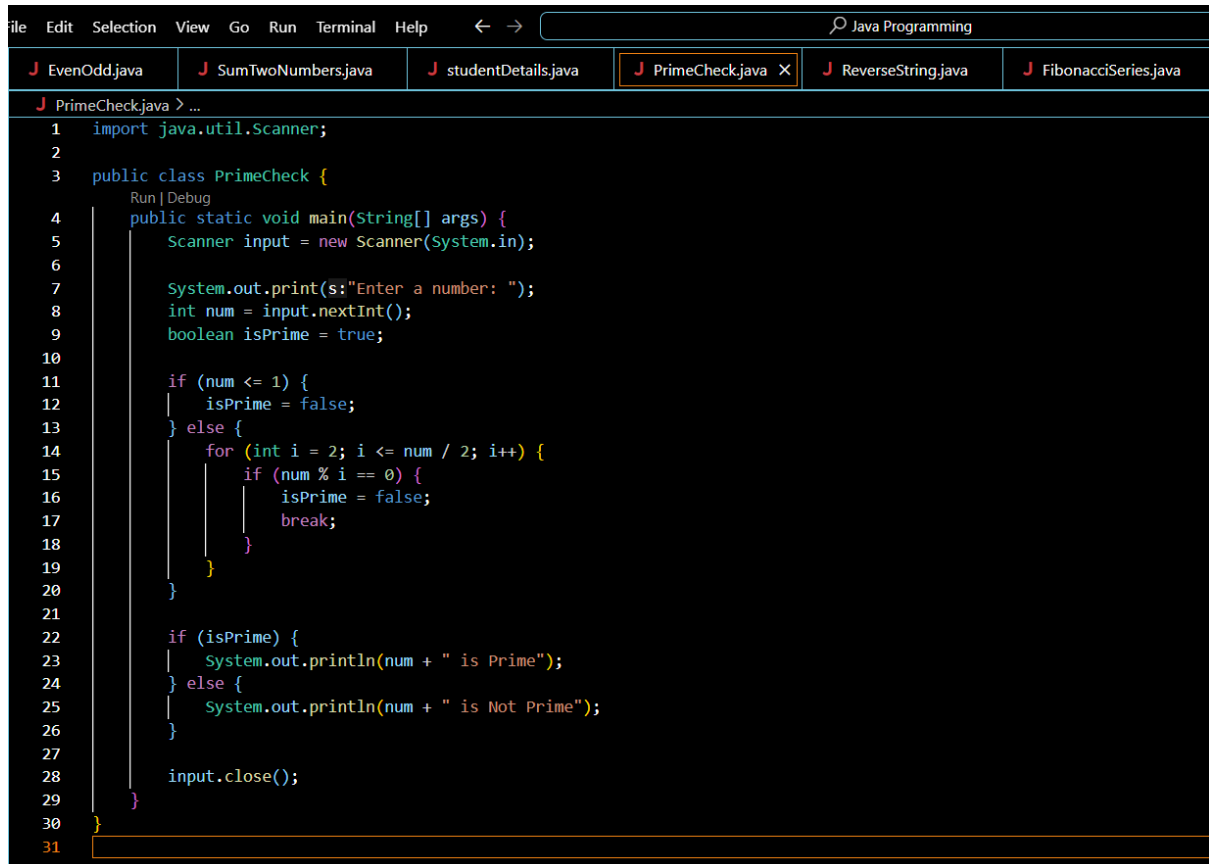
Prints the factorial result in a readable format.

- `input.close();`

Closes the Scanner to release system resources.

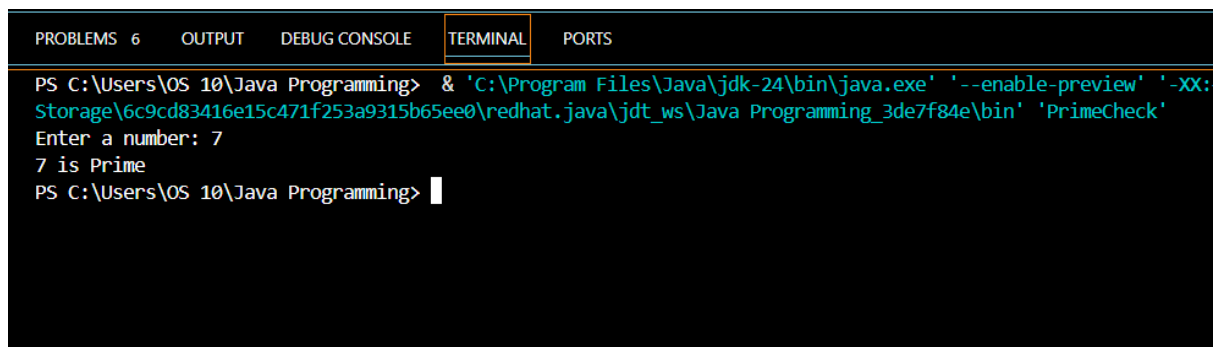
5. Prime Number Check.

Input:



```
1 import java.util.Scanner;
2
3 public class PrimeCheck {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print(s:"Enter a number: ");
8         int num = input.nextInt();
9         boolean isPrime = true;
10
11         if (num <= 1) {
12             isPrime = false;
13         } else {
14             for (int i = 2; i <= num / 2; i++) {
15                 if (num % i == 0) {
16                     isPrime = false;
17                     break;
18                 }
19             }
20         }
21
22         if (isPrime) {
23             System.out.println(num + " is Prime");
24         } else {
25             System.out.println(num + " is Not Prime");
26         }
27
28         input.close();
29     }
30 }
31
```

Output:



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:Storage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\jdt_ws\Java Programming_3de7f84e\bin' 'PrimeCheck'
Enter a number: 7
7 is Prime
PS C:\Users\OS 10\Java Programming>
```

Explanation :

Explanation of Important Lines:

- `boolean isPrime = true;`

Assumes the number is prime at the start.

- `if (num <= 1)`

Immediately disqualifies numbers 1 and below from being prime.

- `for (int i = 2; i <= Math.sqrt(num); i++)`

Efficiently checks possible factors only up to the square root of num because any factor larger than that would have a corresponding smaller factor.

- `if (num % i == 0)`

Checks if num is divisible by i. If yes, it's not prime.

- `isPrime = false; break;`

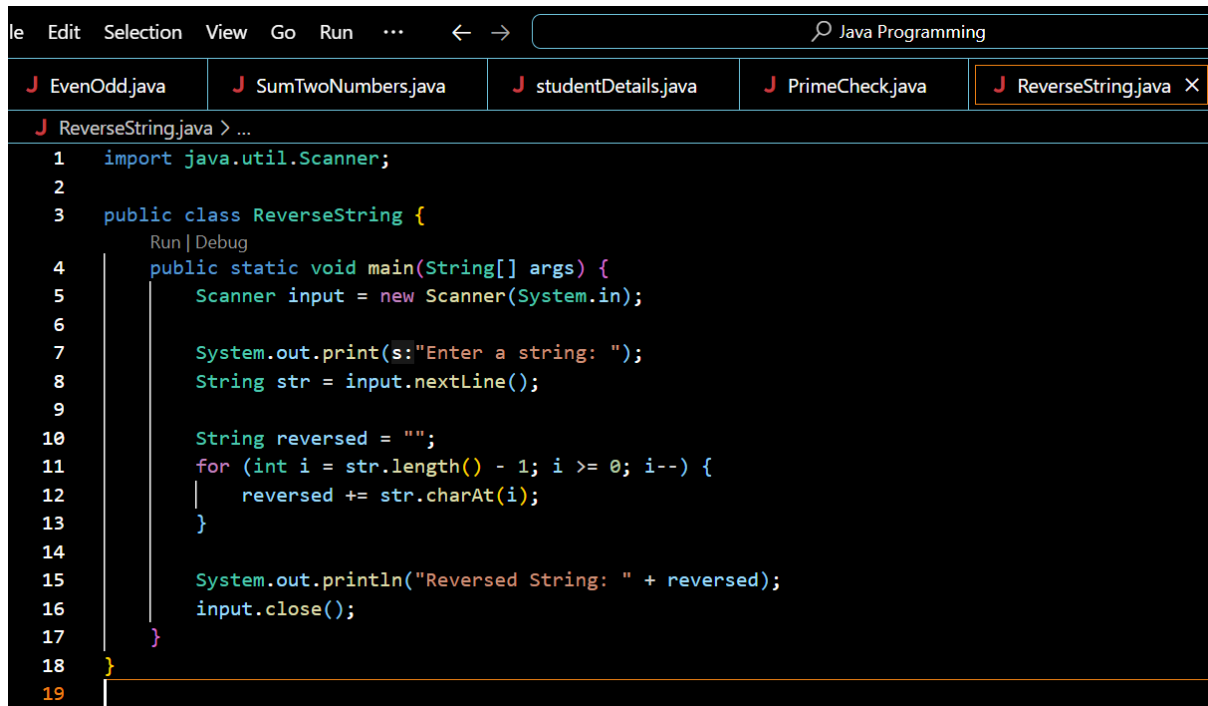
Once a divisor is found, marks num as not prime and stops further checking.

- `if (isPrime) { ... } else { ... }`

Prints the final verdict.

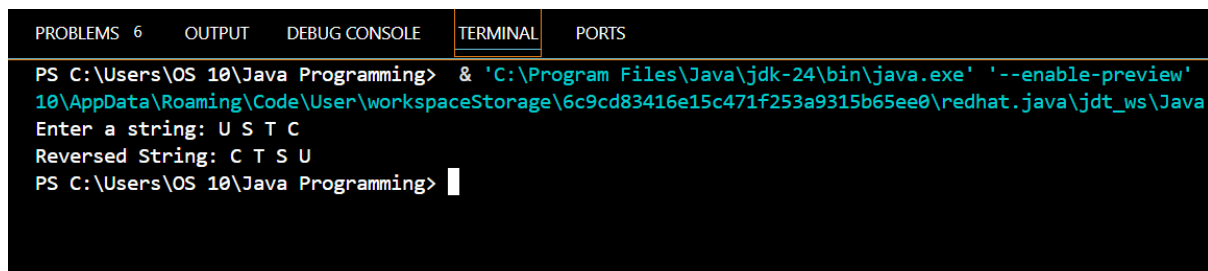
6. Reverse a String

Input:



```
1 import java.util.Scanner;
2
3 public class ReverseString {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print(s:"Enter a string: ");
8         String str = input.nextLine();
9
10        String reversed = "";
11        for (int i = str.length() - 1; i >= 0; i--) {
12            reversed += str.charAt(i);
13        }
14
15        System.out.println("Reversed String: " + reversed);
16        input.close();
17    }
18 }
19
```

Output:



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '
10\AppData\Roaming\Code\User\workspaceStorage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\jdt_ws\Java
Enter a string: U S T C
Reversed String: C T S U
PS C:\Users\OS 10\Java Programming>
```

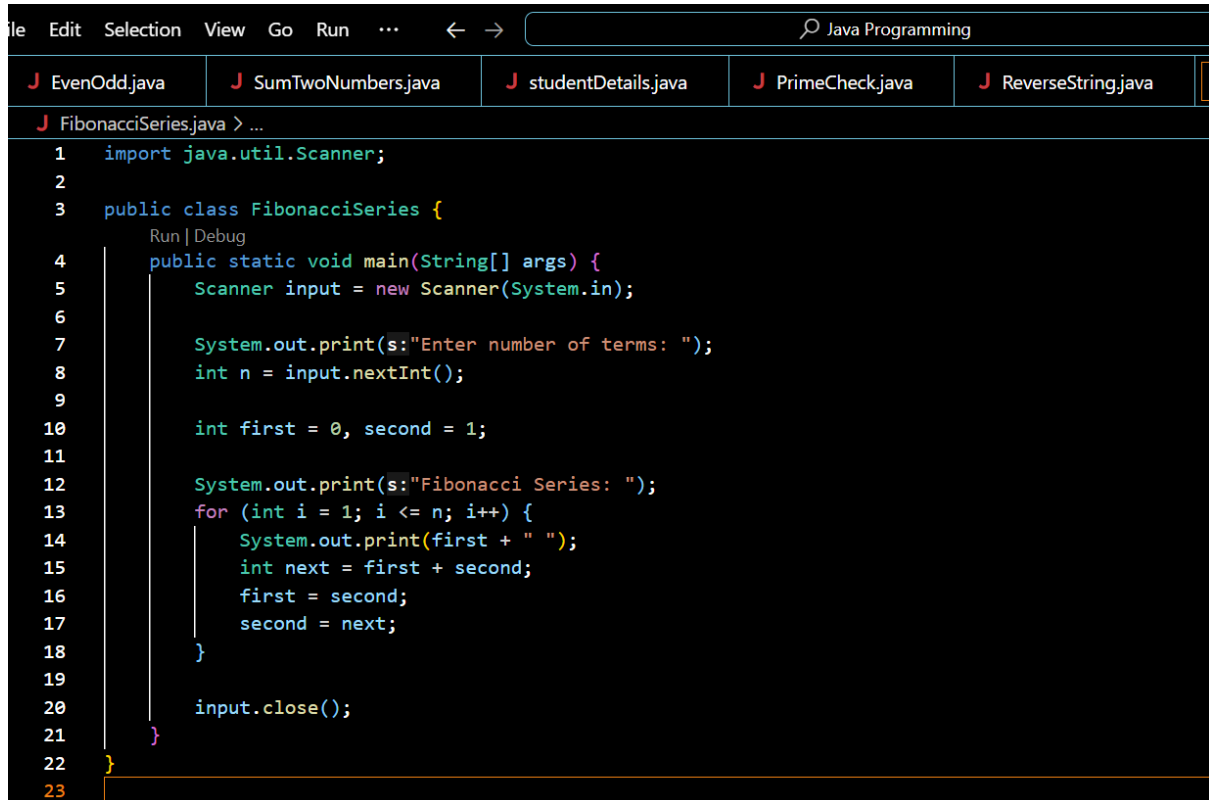
Explanation :

Explanation of Important Lines:

- `String str = input.nextLine();`
Reads the entire line of text input by the user.
- `for (int i = str.length() - 1; i >= 0; i--)`
Loops backwards through the string starting from the last character index to zero.
- `reversed += str.charAt(i);`
Adds each character from the end of `str` to the `reversed` string.
- `System.out.println("Reversed String: " + reversed);`
Prints out the reversed string in a readable format.
- `input.close();`
Closes the Scanner to release system resources.

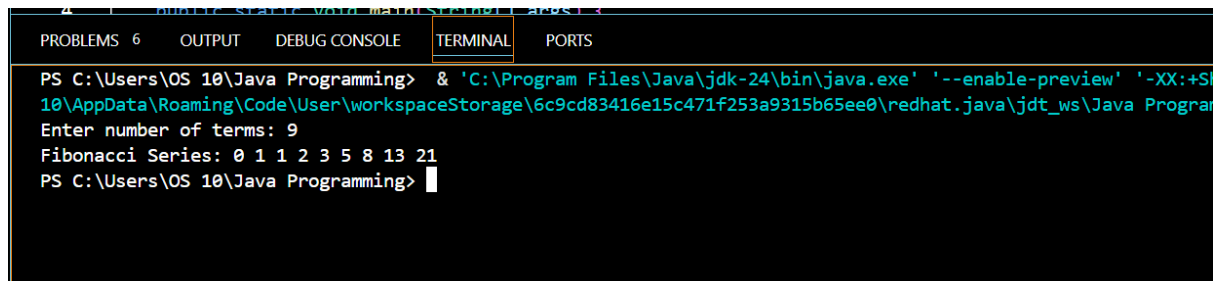
7. Fibonacci Series up to N Terms

Input:



```
1 import java.util.Scanner;
2
3 public class FibonacciSeries {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print(s:"Enter number of terms: ");
8         int n = input.nextInt();
9
10        int first = 0, second = 1;
11
12        System.out.print(s:"Fibonacci Series: ");
13        for (int i = 1; i <= n; i++) {
14            System.out.print(first + " ");
15            int next = first + second;
16            first = second;
17            second = next;
18        }
19
20        input.close();
21    }
22 }
23
```

Output:



```
PS C:\Users\OS 10\Java Programming> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+S
10\AppData\Roaming\Code\User\workspaceStorage\6c9cd83416e15c471f253a9315b65ee0\redhat.java\jdt_ws\Java Program
Enter number of terms: 9
Fibonacci Series: 0 1 1 2 3 5 8 13 21
PS C:\Users\OS 10\Java Programming>
```

Explanation :

First condition: `if (num1 >= num2 && num1 >= num3)` checks if `num1` is greater than or equal to both `num2` and `num3`. If this is true, `num1` is the largest.

Second condition: `else if (num2 >= num1 && num2 >= num3)` is checked only if the first condition is false. It checks if `num2` is greater than or equal to both `num1` and `num3`. If true, `num2` is the largest.

Final condition: `else` is the default case. If neither of the first two conditions is true, it means `num3` must be the largest number.