Assignment-7 for Lambdas and Functional Programming

Subject: CSW2 (CSE 2141)

Name: Arpit Kumar Mohanty

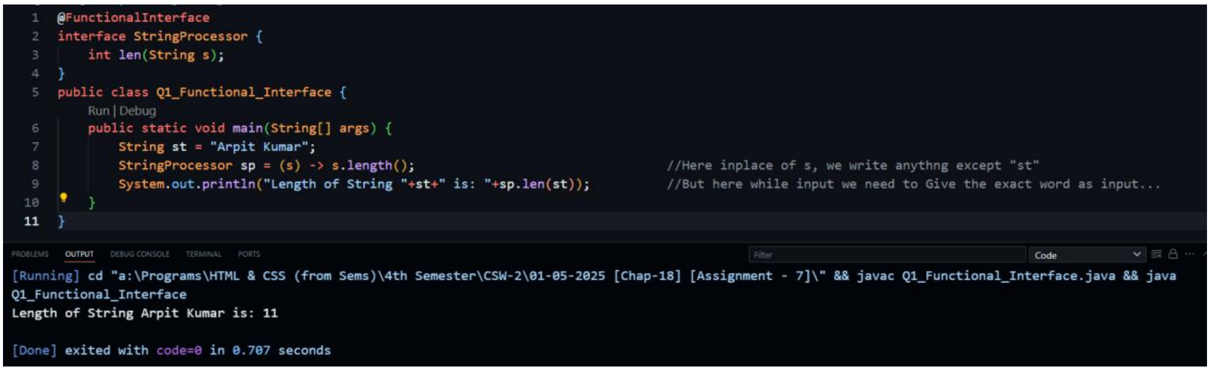
Registration Number: 2341013237

Section: 23412G1

Branch: CSE

Q1. You are required to create a Java program that uses a functional interface to process strings. Define a functional interface named StringProcessor with a method that takes a string as input and returns an integer. Within the main method of a class, use a lambda expression to implement this method such that it returns the length of the given string.

Code along with Output Snippet:

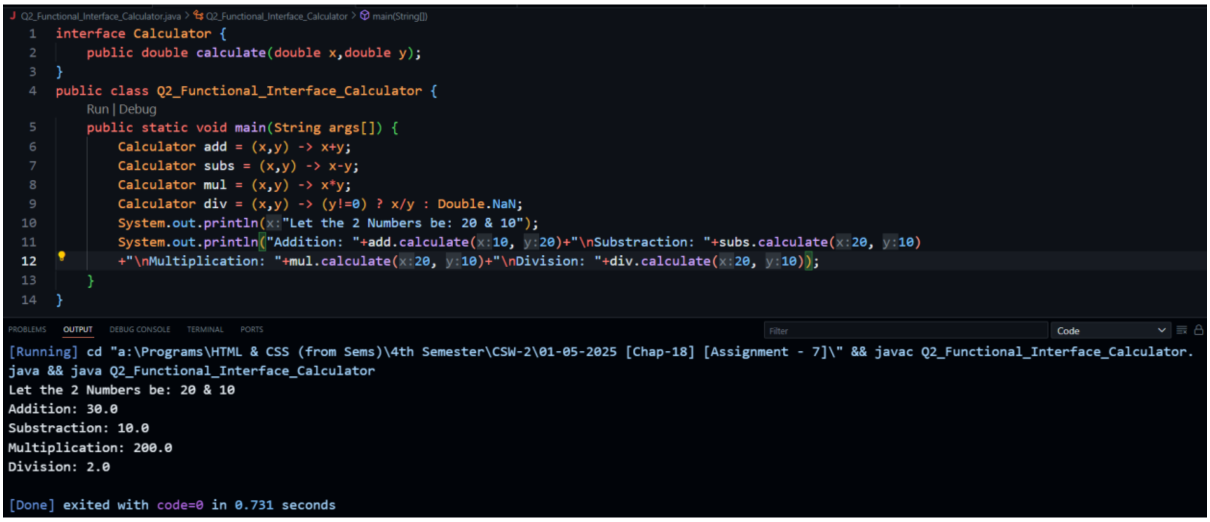


```java
@FunctionalInterface
interface StringProcessor {
    int len(String s);
}
public class Q1_Functional_Interface {
    public static void main(String[] args) {
        String st = "Arpit Kumar";
        StringProcessor sp = (s) -> s.length();                    //Here inplace of s, we write anythng except "st"
        System.out.println("Length of String "+st+" is: "+sp.len(st));       //But here while input we need to Give the exact word as input...
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q1_Functional_Interface.java && java Q1_Functional_Interface
Length of String Arpit Kumar is: 11

[Done] exited with code=0 in 0.707 seconds
```

Q2. Create a functional interface Calculator with methods for addition, subtraction, multiplication, and division. Implement these methods using lambda expressions. Define the Calculator functional interface with methods for arithmetic operations. Implement the interface methods using lambda expressions for basic arithmetic operations.

Code along with Output Snippet:



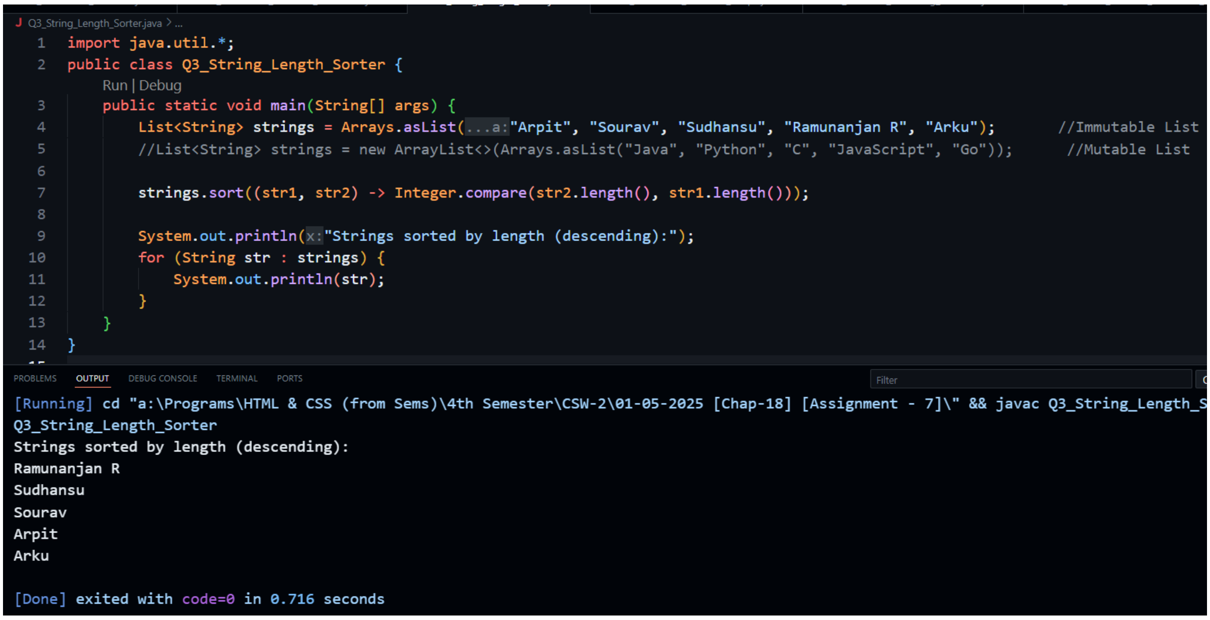
Q2_Functional_Interface_Calculator.java > Q2_Functional_Interface_Calculator > main(String[])
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Filter
Code

```java
interface Calculator {
    public double calculate(double x,double y);
}
public class Q2_Functional_Interface_Calculator {
    Run | Debug
    public static void main(String args[]) {
        Calculator add = (x,y) -> x+y;
        Calculator subs = (x,y) -> x-y;
        Calculator mul = (x,y) -> x*y;
        Calculator div = (x,y) -> (y!=0) ? x/y : Double.NaN;
        System.out.println(x:"Let the 2 Numbers be: 20 & 10");
        System.out.println("Addition: "+add.calculate(x:10, y:20)+"\nSubstraction: "+subs.calculate(x:20, y:10)
        +"\nMultiplication: "+mul.calculate(x:20, y:10)+"\nDivision: "+div.calculate(x:20, y:10));
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q2_Functional_Interface_Calculator.
java && java Q2_Functional_Interface_Calculator
Let the 2 Numbers be: 20 & 10
Addition: 30.0
Substraction: 10.0
Multiplication: 200.0
Division: 2.0

[Done] exited with code=0 in 0.731 seconds
```

Q3. Write a program that sorts a list of strings based on their lengths in descending order. Define a custom comparator using a lambda expression that compares strings based on their lengths. Use the custom comparator to sort the list of strings in descending order of length.

Code along with Output Snippet:

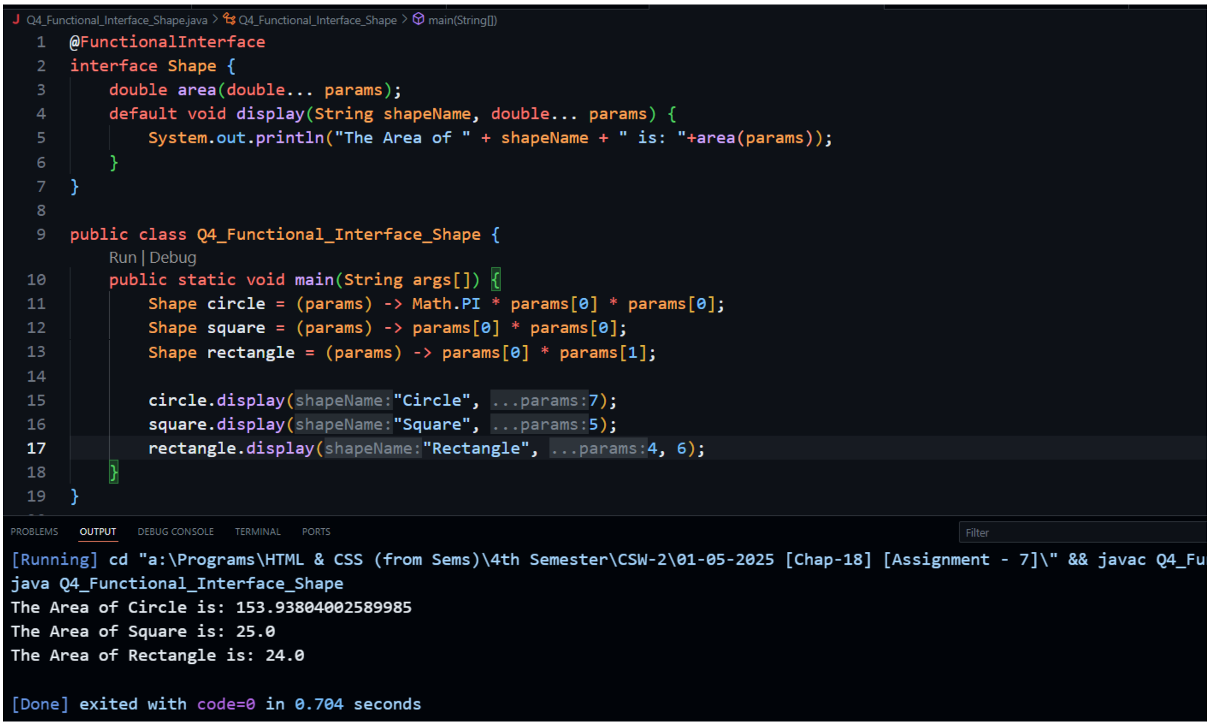


```java
import java.util.*;
public class Q3_String_Length_Sorter {
    Run | Debug
    public static void main(String[] args) {
        List<String> strings = Arrays.asList(...a:"Arpit", "Sourav", "Sudhansu", "Ramunanjan R", "Arku");      //Immutable List
        //List<String> strings = new ArrayList<>(Arrays.asList("Java", "Python", "C", "JavaScript", "Go"));      //Mutable List

        strings.sort((str1, str2) -> Integer.compare(str2.length(), str1.length()));

        System.out.println(x:"Strings sorted by length (descending):");
        for (String str : strings) {
            System.out.println(str);
        }
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q3_String_Length_S
Q3_String_Length_Sorter
Strings sorted by length (descending):
Ramunanjan R
Sudhansu
Sourav
Arpit
Arku

[Done] exited with code=0 in 0.716 seconds
```

Q4. Create a functional interface Shape with a method double area() and a default method void printArea(). Implement the interface using lambda expressions for different shapes. Define the Shape functional interface with an area method. Implement the interface for shapes like circle, square, and rectangle using lambda expressions. Use the default method to print the area of each shape.

Code along with Output Snippet:

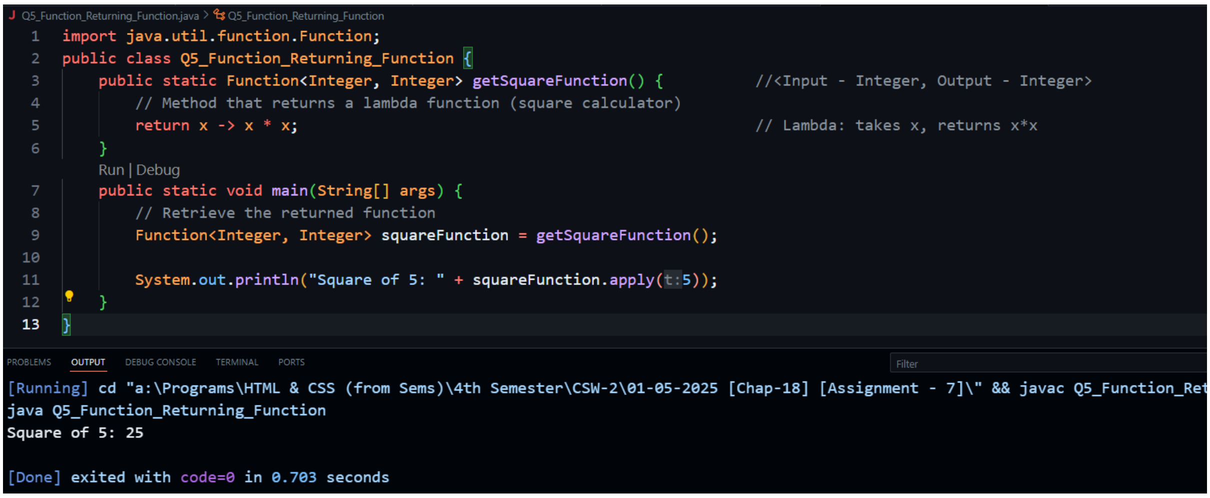

Q4_Functional_Interface_Shape.java > Q4_Functional_Interface_Shape > main(String[])
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Filter

```java
@FunctionalInterface
interface Shape {
    double area(double... params);
    default void display(String shapeName, double... params) {
        System.out.println("The Area of " + shapeName + " is: "+area(params));
    }
}

public class Q4_Functional_Interface_Shape {
    Run | Debug
    public static void main(String args[]) {
        Shape circle = (params) -> Math.PI * params[0] * params[0];
        Shape square = (params) -> params[0] * params[0];
        Shape rectangle = (params) -> params[0] * params[1];

        circle.display(shapeName:"Circle", ...params:7);
        square.display(shapeName:"Square", ...params:5);
        rectangle.display(shapeName:"Rectangle", ...params:4, 6);
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q4_Fu
java Q4_Functional_Interface_Shape
The Area of Circle is: 153.93804002589985
The Area of Square is: 25.0
The Area of Rectangle is: 24.0

[Done] exited with code=0 in 0.704 seconds
```

Q5. Write a Java program that demonstrates the concept of a function returning another function using Java's Function interface. Define a method that returns a lambda function that takes an integer as input and returns its square. In the main method, retrieve the returned function and use it to calculate and display the squares of different numbers.

Code along with Output Snippet:

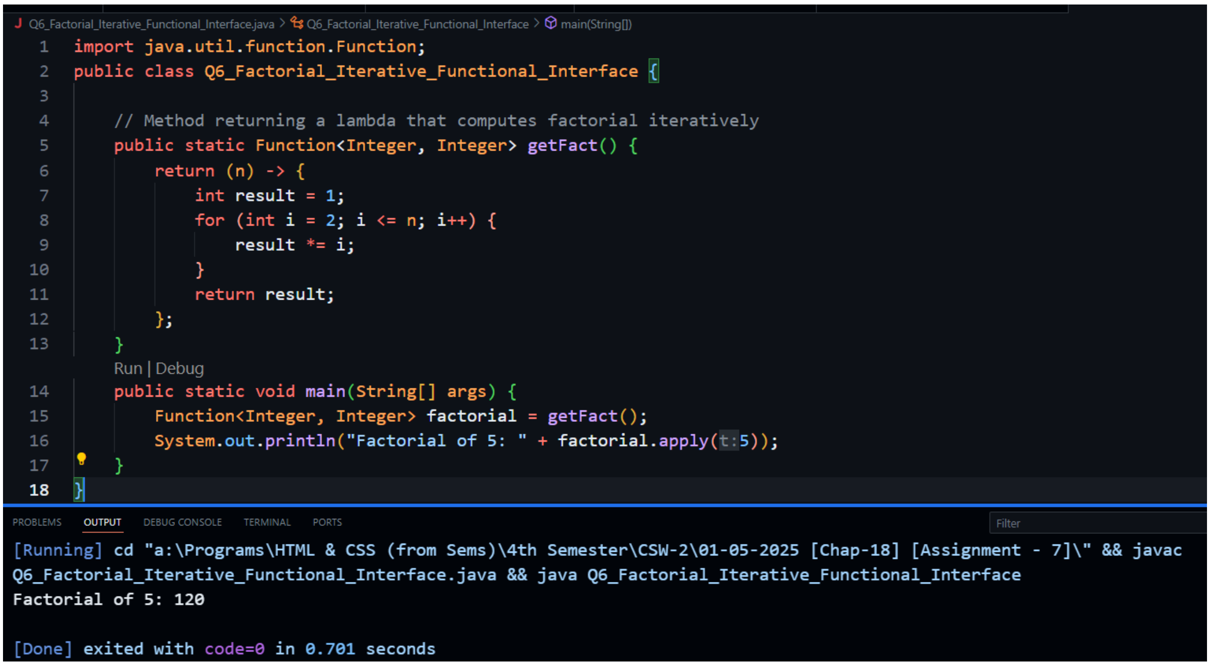

Q5_Function_Returning_Function.java > Q5_Function_Returning_Function
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Filter

```java
import java.util.function.Function;
public class Q5_Function_Returning_Function {
    public static Function<Integer, Integer> getSquareFunction() {        //<Input - Integer, Output - Integer>
        // Method that returns a lambda function (square calculator)
        return x -> x * x;                                                 // Lambda: takes x, returns x*x
    }
    Run | Debug
    public static void main(String[] args) {
        // Retrieve the returned function
        Function<Integer, Integer> squareFunction = getSquareFunction();

        System.out.println("Square of 5: " + squareFunction.apply(t:5));
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q5_Function_Ret
java Q5_Function_Returning_Function
Square of 5: 25

[Done] exited with code=0 in 0.703 seconds
```

Q6. Write a Java program that demonstrates the use of a lambda expression with the Function functional interface to calculate the factorial of a number. Define a method that returns a lambda expression that takes an integer as input and computes its factorial using an iterative approach. In the main method, use this lambda expression to calculate and display the factorial of a given number

Code along with Output Snippet:


Q6_Factorial_Iterative_Functional_Interface.java > Q6_Factorial_Iterative_Functional_Interface > main(String[])
Run | Debug
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Filter

```java
import java.util.function.Function;
public class Q6_Factorial_Iterative_Functional_Interface {

    // Method returning a lambda that computes factorial iteratively
    public static Function<Integer, Integer> getFact() {
        return (n) -> {
            int result = 1;
            for (int i = 2; i <= n; i++) {
                result *= i;
            }
            return result;
        };
    }
    public static void main(String[] args) {
        Function<Integer, Integer> factorial = getFact();
        System.out.println("Factorial of 5: " + factorial.apply(t:5));
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac
Q6_Factorial_Iterative_Functional_Interface.java && java Q6_Factorial_Iterative_Functional_Interface
Factorial of 5: 120

[Done] exited with code=0 in 0.701 seconds
```

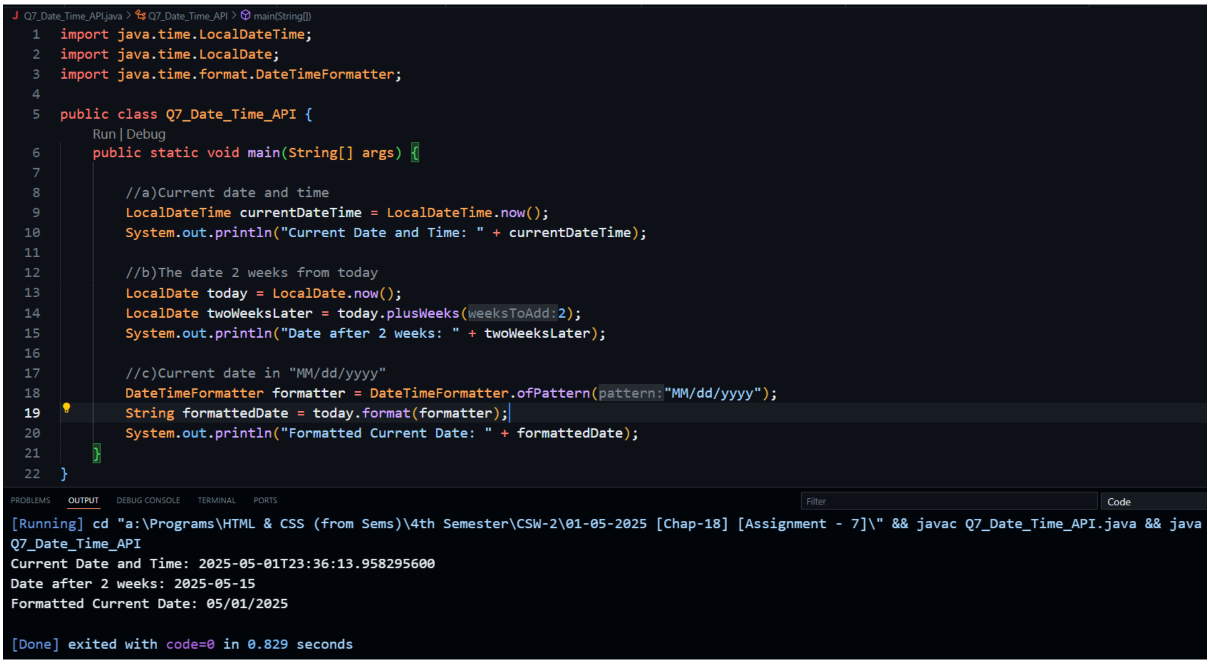
Q7. Using the Java Date and Time API, write a Java program to perform the following tasks:

a) Retrieve and display the current date and time.

b) Calculate the date that is 2 weeks from today.

c) Format and display the current date in the "MM/dd/yyyy" format.

Code along with Output Snippet:

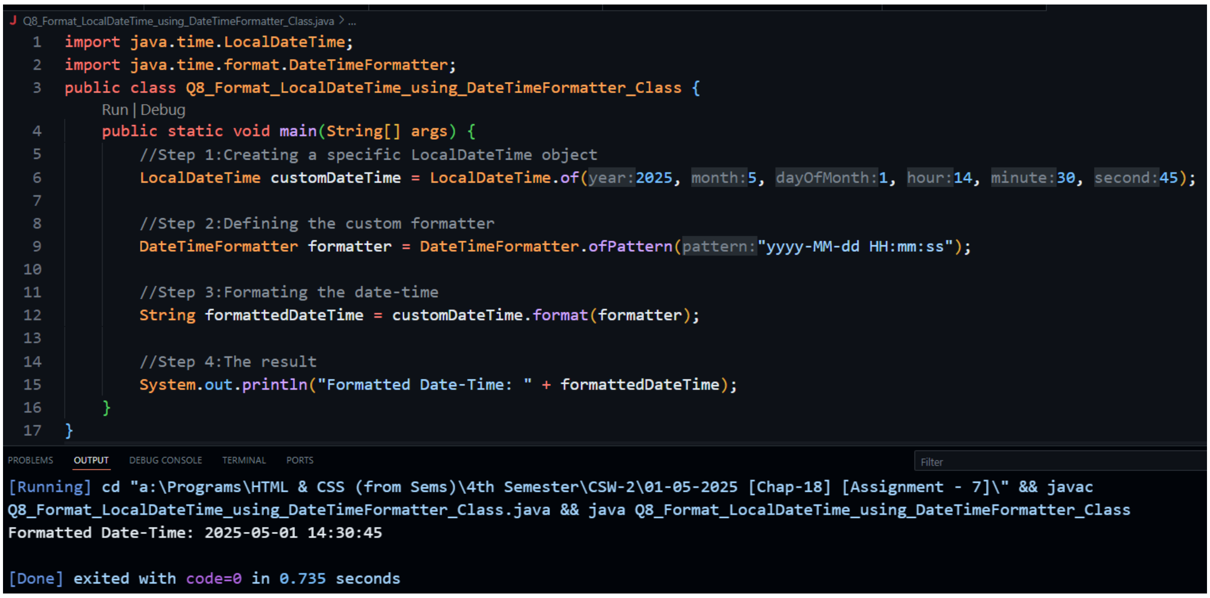


```java
import java.time.LocalDateTime;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class Q7_Date_Time_API {
    public static void main(String[] args) {

        //a)Current date and time
        LocalDateTime currentDateTime = LocalDateTime.now();
        System.out.println("Current Date and Time: " + currentDateTime);

        //b)The date 2 weeks from today
        LocalDate today = LocalDate.now();
        LocalDate twoWeeksLater = today.plusWeeks(weeksToAdd:2);
        System.out.println("Date after 2 weeks: " + twoWeeksLater);

        //c)Current date in "MM/dd/yyyy"
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern:"MM/dd/yyyy");
        String formattedDate = today.format(formatter);
        System.out.println("Formatted Current Date: " + formattedDate);
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac Q7_Date_Time_API.java && java
Q7_Date_Time_API
Current Date and Time: 2025-05-01T23:36:13.958295600
Date after 2 weeks: 2025-05-15
Formatted Current Date: 05/01/2025

[Done] exited with code=0 in 0.829 seconds
```

Q8. Write a Java program that demonstrates how to format a LocalDateTime object using the DateTimeFormatter class. Create a LocalDateTime instance representing a specific date and time. Then, use a custom format pattern "yyyy-MM-dd HH:mm:ss" to convert this date-time into a formatted string. Finally, print the formatted result.

Code along with Output Snippet:



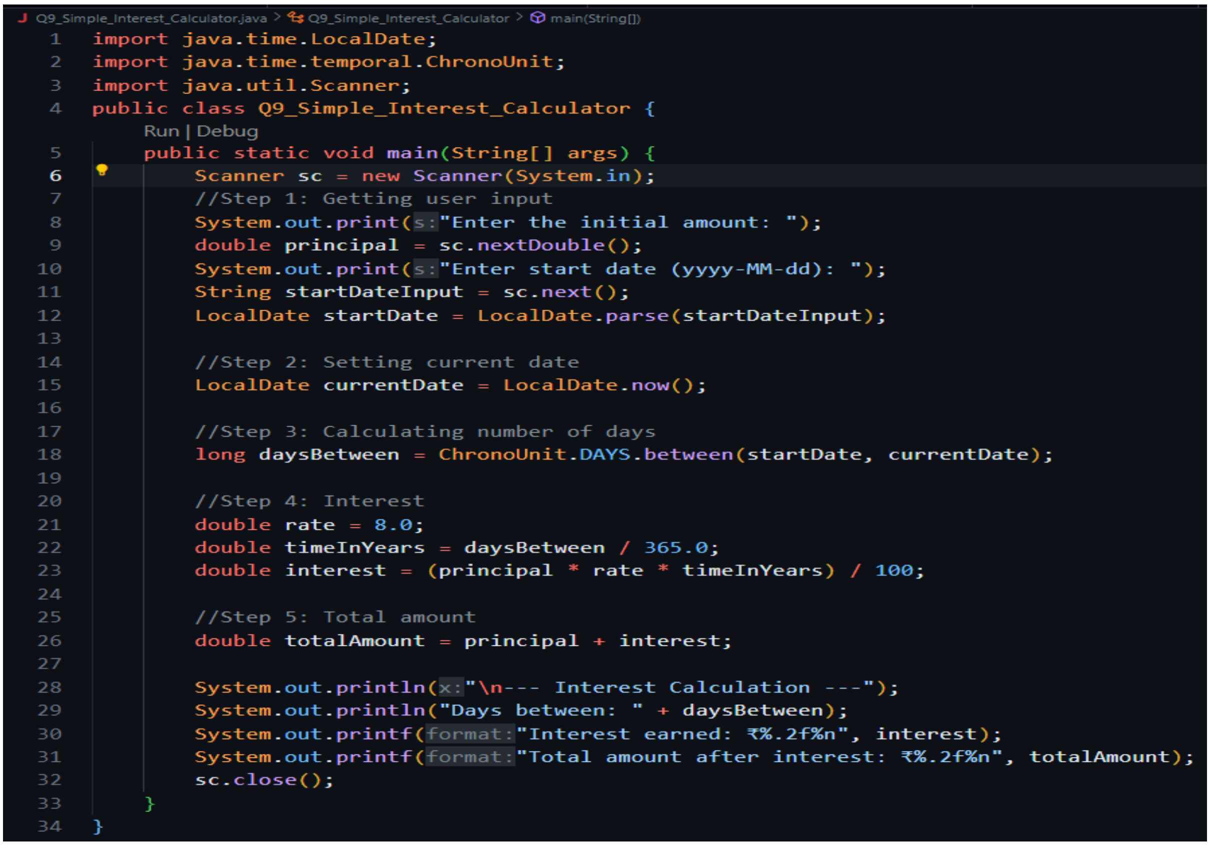
Q8_Format_LocalDateTime_using_DateTimeFormatter_Class.java > ...

```java
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
public class Q8_Format_LocalDateTime_using_DateTimeFormatter_Class {
    Run | Debug
    public static void main(String[] args) {
        //Step 1:Creating a specific LocalDateTime object
        LocalDateTime customDateTime = LocalDateTime.of(year:2025, month:5, dayOfMonth:1, hour:14, minute:30, second:45);

        //Step 2:Defining the custom formatter
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern:"yyyy-MM-dd HH:mm:ss");

        //Step 3:Formating the date-time
        String formattedDateTime = customDateTime.format(formatter);

        //Step 4:The result
        System.out.println("Formatted Date-Time: " + formattedDateTime);
    }
}
```

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\01-05-2025 [Chap-18] [Assignment - 7]\" && javac
Q8_Format_LocalDateTime_using_DateTimeFormatter_Class.java && java Q8_Format_LocalDateTime_using_DateTimeFormatter_Class
Formatted Date-Time: 2025-05-01 14:30:45

[Done] exited with code=0 in 0.735 seconds
```

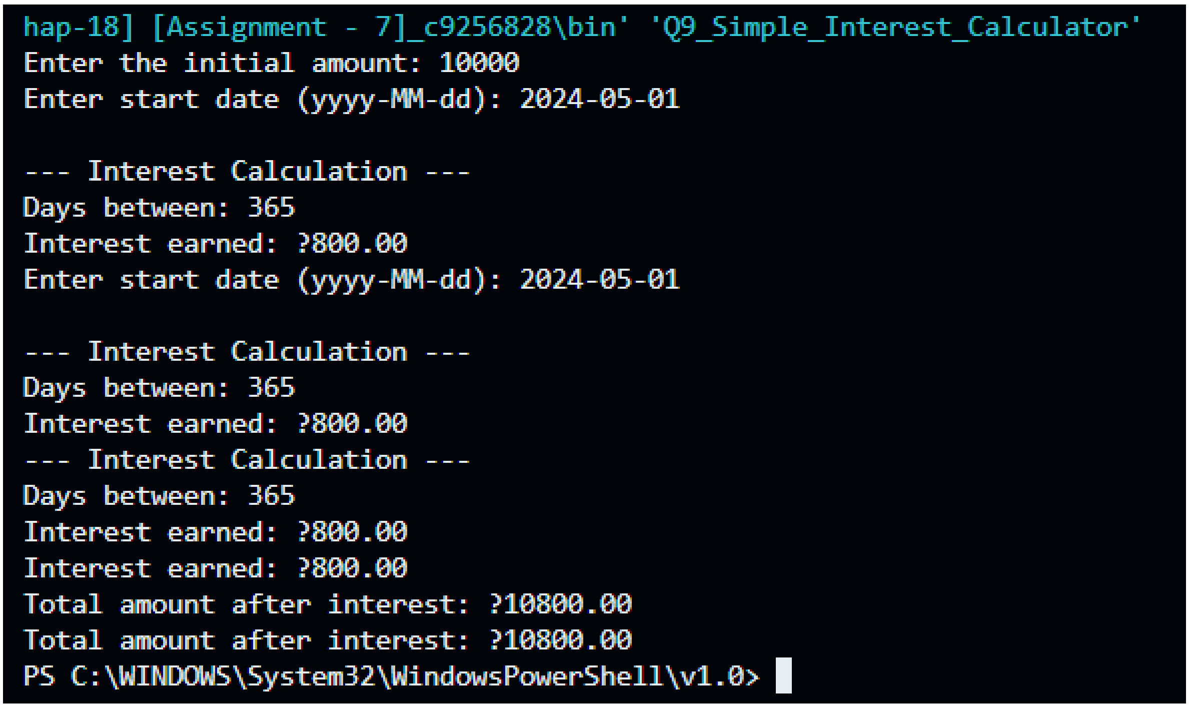
Q9. Write a Java program to calculate the simple interest on a current account using the Java Date and Time API. The program should prompt the user to enter the start date and the initial amount. Assume the rate of interest is 8% per annum. Calculate the number of days between the start date and the current date, then compute the interest using the simple interest formula. Finally, display the total amount after adding the interest to the initial amount.

Code along with Output Snippet:



Q9_Simple_Interest_Calculator.java > Q9_Simple_Interest_Calculator > main(String[])

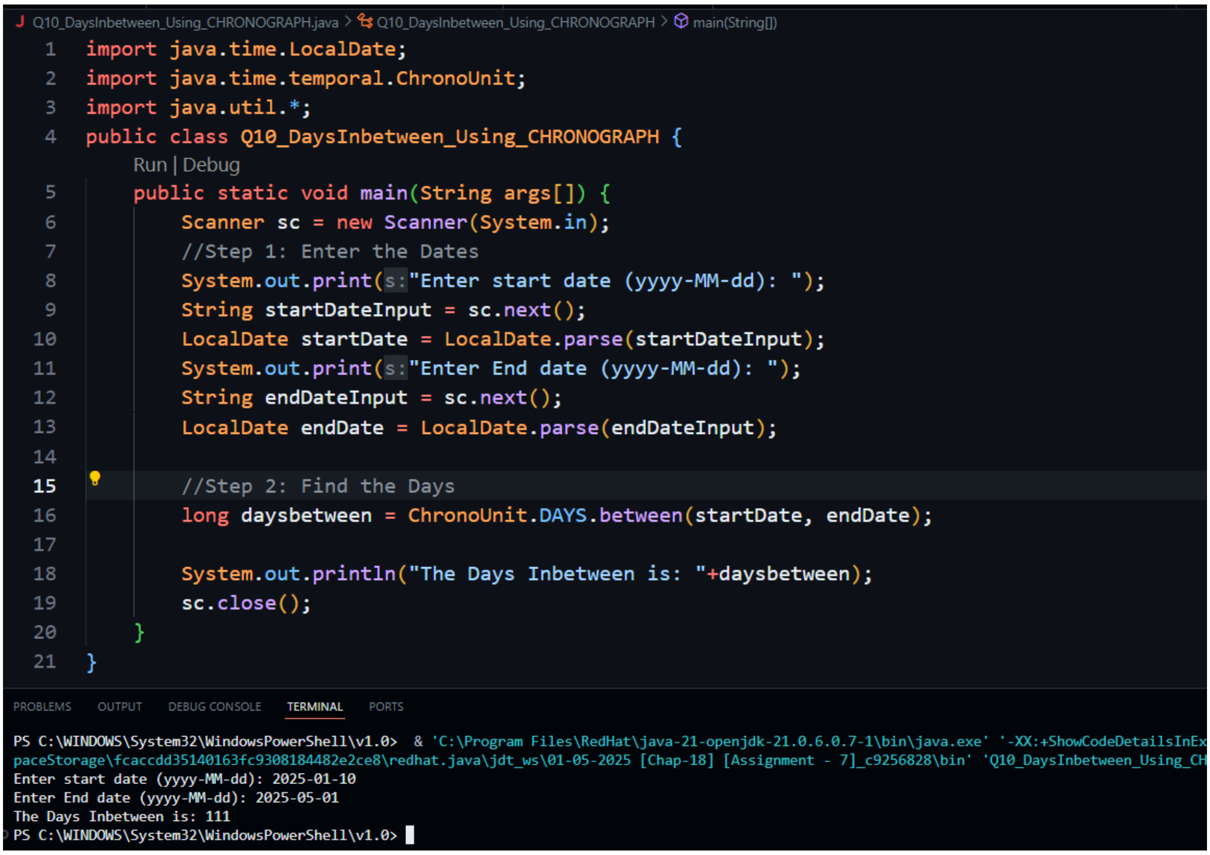
```java
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;
public class Q9_Simple_Interest_Calculator {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Step 1: Getting user input
        System.out.print(s:"Enter the initial amount: ");
        double principal = sc.nextDouble();
        System.out.print(s:"Enter start date (yyyy-MM-dd): ");
        String startDateInput = sc.next();
        LocalDate startDate = LocalDate.parse(startDateInput);

        //Step 2: Setting current date
        LocalDate currentDate = LocalDate.now();

        //Step 3: Calculating number of days
        long daysBetween = ChronoUnit.DAYS.between(startDate, currentDate);

        //Step 4: Interest
        double rate = 8.0;
        double timeInYears = daysBetween / 365.0;
        double interest = (principal * rate * timeInYears) / 100;

        //Step 5: Total amount
        double totalAmount = principal + interest;

        System.out.println(x:"\n--- Interest Calculation ---");
        System.out.println("Days between: " + daysBetween);
        System.out.printf(format:"Interest earned: ₹%.2f%n", interest);
        System.out.printf(format:"Total amount after interest: ₹%.2f%n", totalAmount);
        sc.close();
    }
}
```



```
hap-18] [Assignment - 7]_c9256828\bin' 'Q9_Simple_Interest_Calculator'
Enter the initial amount: 10000
Enter start date (yyyy-MM-dd): 2024-05-01

--- Interest Calculation ---
Days between: 365
Interest earned: ?800.00
Enter start date (yyyy-MM-dd): 2024-05-01

--- Interest Calculation ---
Days between: 365
Interest earned: ?800.00
--- Interest Calculation ---
Days between: 365
Interest earned: ?800.00
Interest earned: ?800.00
Total amount after interest: ?10800.00
Total amount after interest: ?10800.00
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q10. Write a Java program using the Java Date and Time API to calculate the number of days between two dates. The program should prompt the user to enter a start date and an end date (in YYYY-MM-DD format), and then display the total number of days between the two dates using the ChronoUnit.DAYS method.

Code along with Output Snippet:


Q10_DaysInbetween_Using_CHRONOGRAPH.java > Q10_DaysInbetween_Using_CHRONOGRAPH > main(String[])
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```java
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
import java.util.*;
public class Q10_DaysInbetween_Using_CHRONOGRAPH {
    Run | Debug
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        //Step 1: Enter the Dates
        System.out.print(s:"Enter start date (yyyy-MM-dd): ");
        String startDateInput = sc.next();
        LocalDate startDate = LocalDate.parse(startDateInput);
        System.out.print(s:"Enter End date (yyyy-MM-dd): ");
        String endDateInput = sc.next();
        LocalDate endDate = LocalDate.parse(endDateInput);

        //Step 2: Find the Days
        long daysbetween = ChronoUnit.DAYS.between(startDate, endDate);

        System.out.println("The Days Inbetween is: "+daysbetween);
        sc.close();
    }
}
```

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21.0.6.0.7-1\bin\java.exe' '-XX:+ShowCodeDetailsInEx
paceStorage\fcaccdd35140163fc9308184482e2ce8\redhat.java\jdt_ws\01-05-2025 [Chap-18] [Assignment - 7]_c9256828\bin' 'Q10_DaysInbetween_Using_CH
Enter start date (yyyy-MM-dd): 2025-01-10
Enter End date (yyyy-MM-dd): 2025-05-01
The Days Inbetween is: 111
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```