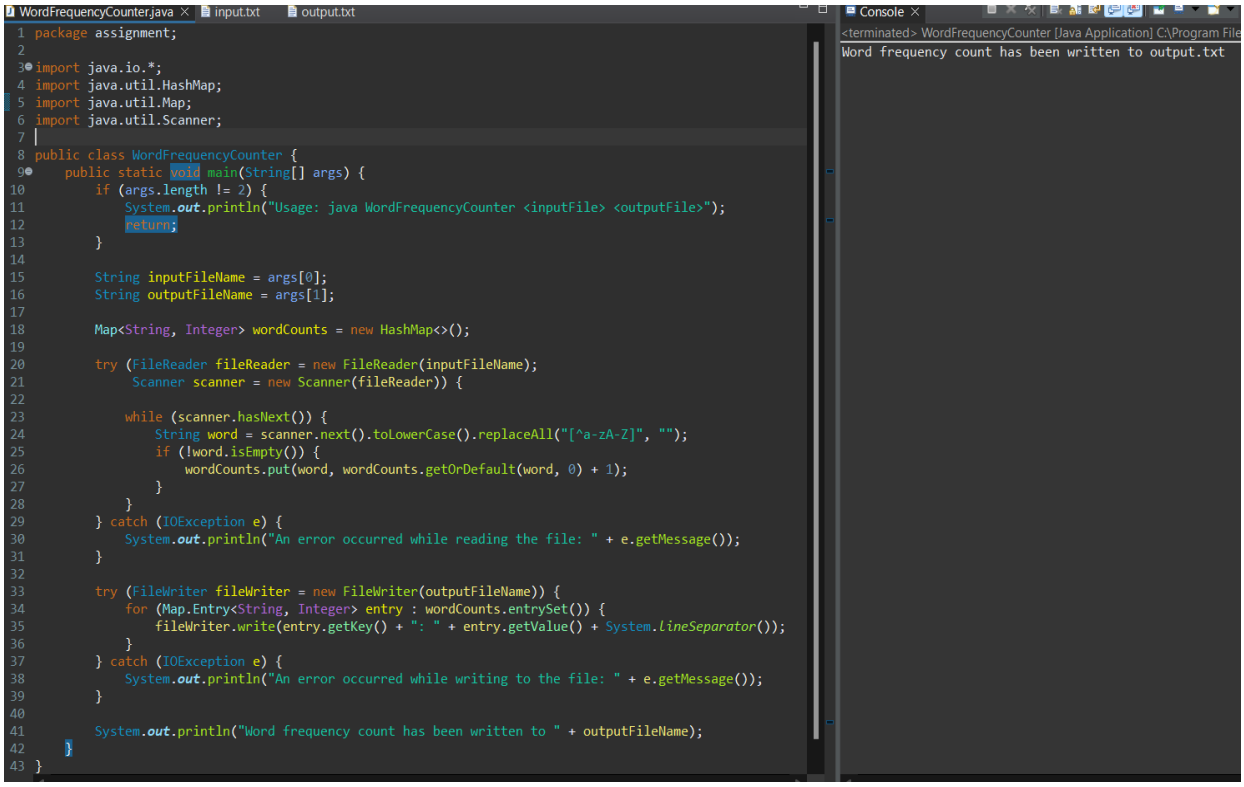
DAY-20 JAVA ASSIGNMENT

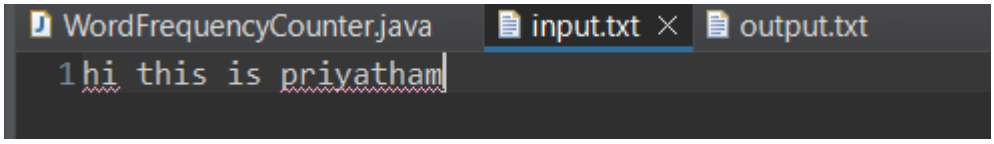
Task 1: Java IO Basics

Write a program that reads a text file and counts the frequency of each word using FileReader and FileWriter.

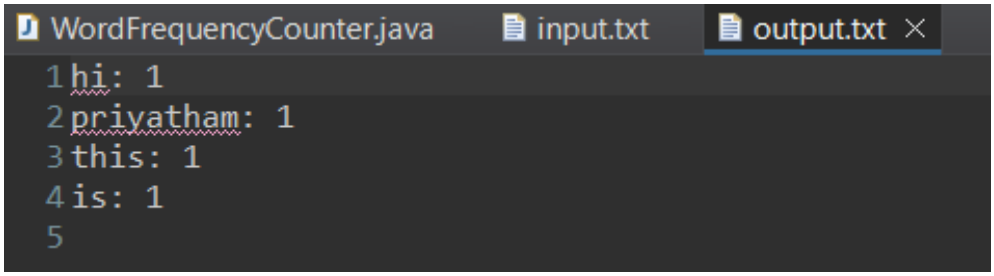


```java
package assignment;

import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class WordFrequencyCounter {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java WordFrequencyCounter <inputFile> <outputFile>");
            return;
        }

        String inputFileName = args[0];
        String outputFileName = args[1];

        Map<String, Integer> wordCounts = new HashMap<>();

        try (FileReader fileReader = new FileReader(inputFileName);
             Scanner scanner = new Scanner(fileReader)) {

            while (scanner.hasNext()) {
                String word = scanner.next().toLowerCase().replaceAll("[^a-zA-Z]", "");
                if (!word.isEmpty()) {
                    wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);
                }
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file: " + e.getMessage());
        }

        try (FileWriter fileWriter = new FileWriter(outputFileName)) {
            for (Map.Entry<String, Integer> entry : wordCounts.entrySet()) {
                fileWriter.write(entry.getKey() + ": " + entry.getValue() + System.lineSeparator());
            }
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the file: " + e.getMessage());
        }

        System.out.println("Word frequency count has been written to " + outputFileName);
    }
}
```

```
Console
<terminated> WordFrequencyCounter [Java Application] C:\Program File
Word frequency count has been written to output.txt
```



```
WordFrequencyCounter.java   input.txt   output.txt
1 hi this is priyatham
```



```
WordFrequencyCounter.java   input.txt   output.txt
1 hi: 1
2 priyatham: 1
3 this: 1
4 is: 1
5
```

Task 2: Serialization and Deserialization

Serialize a custom object to a file and then deserialize it back to recover the object state.

SerialDeserial.java

```java
package assignment;

import java.io.*;
class Student implements Serializable {
    private String name;
    private int age;
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
}
public class SerialDeserial {
    public static void main(String[] args) {
        Student student = new Student("Priyatham", 23);
        serializeObject(student, "student.ser");
        Student deserializedStudent = (Student) deserializeObject("student.ser");
        System.out.println("Deserialized Student:");
        System.out.println("Name: " + deserializedStudent.getName());
        System.out.println("Age: " + deserializedStudent.getAge());
    }

    public static void serializeObject(Object obj, String fileName) {
        try {
            FileOutputStream fileOut = new FileOutputStream(fileName);
            ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);
            objectOut.writeObject(obj);
            objectOut.close();
            fileOut.close();
            System.out.println("Object serialized successfully.");
        } catch (IOException e) {
            System.err.println("Error during serialization: " + e.getMessage());
        }
    }
```

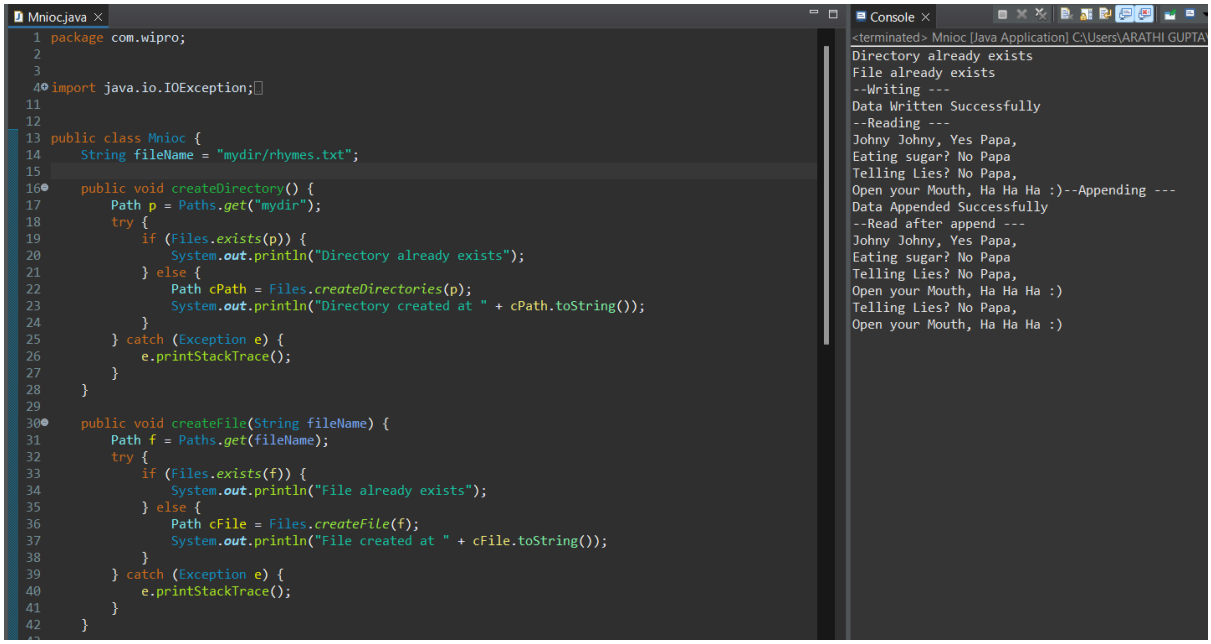
Console

```
<terminated> SerialDeserial [Java Application]
Object serialized successfully.
Object deserialized successfully.
Deserialized Student:
Name: Priyatham
Age: 23
```

```java
    public static Object deserializeObject(String fileName) {
        try {
            FileInputStream fileIn = new FileInputStream(fileName);
            ObjectInputStream objectIn = new ObjectInputStream(fileIn);
            Object obj = objectIn.readObject();
            objectIn.close();
            fileIn.close();
            System.out.println("Object deserialized successfully.");
            return obj;
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error during deserialization: " + e.getMessage());
            return null;
        }
    }
}
```

Task 3: New IO (NIO)

Use NIO Channels and Buffers to read content from a file and write to another file.



```java
package com.wipro;


import java.io.IOException;


public class Mnioc {
    String fileName = "mydir/rhymes.txt";

    public void createDirectory() {
        Path p = Paths.get("mydir");
        try {
            if (Files.exists(p)) {
                System.out.println("Directory already exists");
            } else {
                Path cPath = Files.createDirectories(p);
                System.out.println("Directory created at " + cPath.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void createFile(String fileName) {
        Path f = Paths.get(fileName);
        try {
            if (Files.exists(f)) {
                System.out.println("File already exists");
            } else {
                Path cFile = Files.createFile(f);
                System.out.println("File created at " + cFile.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
<terminated> Mnioc [Java Application] C:\Users\ARATHI GUPTA\
Directory already exists
File already exists
--Writing ---
Data Written Successfully
--Reading ---
Johny Johny, Yes Papa,
Eating sugar? No Papa
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)--Appending ---
Data Appended Successfully
--Read after append ---
Johny Johny, Yes Papa,
Eating sugar? No Papa
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
```

```java
    public void readFile() {
        Path f = Paths.get(fileName);
        try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.READ)) {
            ByteBuffer buffer = ByteBuffer.allocate(1024);
            while (fileChannel.read(buffer) > 0) {
                buffer.flip();
                while (buffer.hasRemaining()) {
                    System.out.print((char) buffer.get());
                }
                buffer.clear();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void writeFile(String fileName) {
        Path f = Paths.get(fileName);
        try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.WRITE, StandardOpenOption.CREATE)) {
            String content = "Johny Johny, Yes Papa,\nEating sugar? No Papa";
            ByteBuffer buffer = ByteBuffer.wrap(content.getBytes());
            fileChannel.write(buffer);
            System.out.println("Data Written Successfully");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void appendFile(String fileName) {
        Path f = Paths.get(fileName);
        try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.APPEND)) {
            String content = "\nTelling Lies? No Papa,\nOpen your Mouth, Ha Ha Ha :)";
            ByteBuffer buffer = ByteBuffer.wrap(content.getBytes());
            fileChannel.write(buffer);
            System.out.println("Data Appended Successfully");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

```

```
    public static void main(String[] args) {
        Mnioc mn = new Mnioc();

        // Create a directory
        mn.createDirectory();

        // Create a file
        mn.createFile("mydir/rhymes.txt");
        System.out.println("--Writing ---");
        // Write to a file
        mn.writeFile(mn.fileName);
        System.out.println("--Reading ---");
        // Read from file
        mn.readFile();
        System.out.println("--Appending ---");
        // Append to a file
        mn.appendFile(mn.fileName);
        System.out.println("--Read after append ---");
        // Read from file
        mn.readFile();
    }
}
```

Task 4: Java Networking

Write a simple HTTP client that connects to a URL, sends a request, and displays the response headers and body.

JavaNetworking.java

```java
package assignment;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class JavaNetworking {
    public static void main(String[] args) {
        String urlStr = "http://example.com";

        try {
        URL url = new URL(urlStr);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        int responseCode = conn.getResponseCode();
        System.out.println("Response Code: " + responseCode);
        System.out.println("Response Headers:");
        conn.getHeaderFields().forEach((key, value) -> {
                System.out.println(key + ": " + value);
            });
        System.out.println("Response Body:");
        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        String line;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
         }
         reader.close();
         conn.disconnect();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

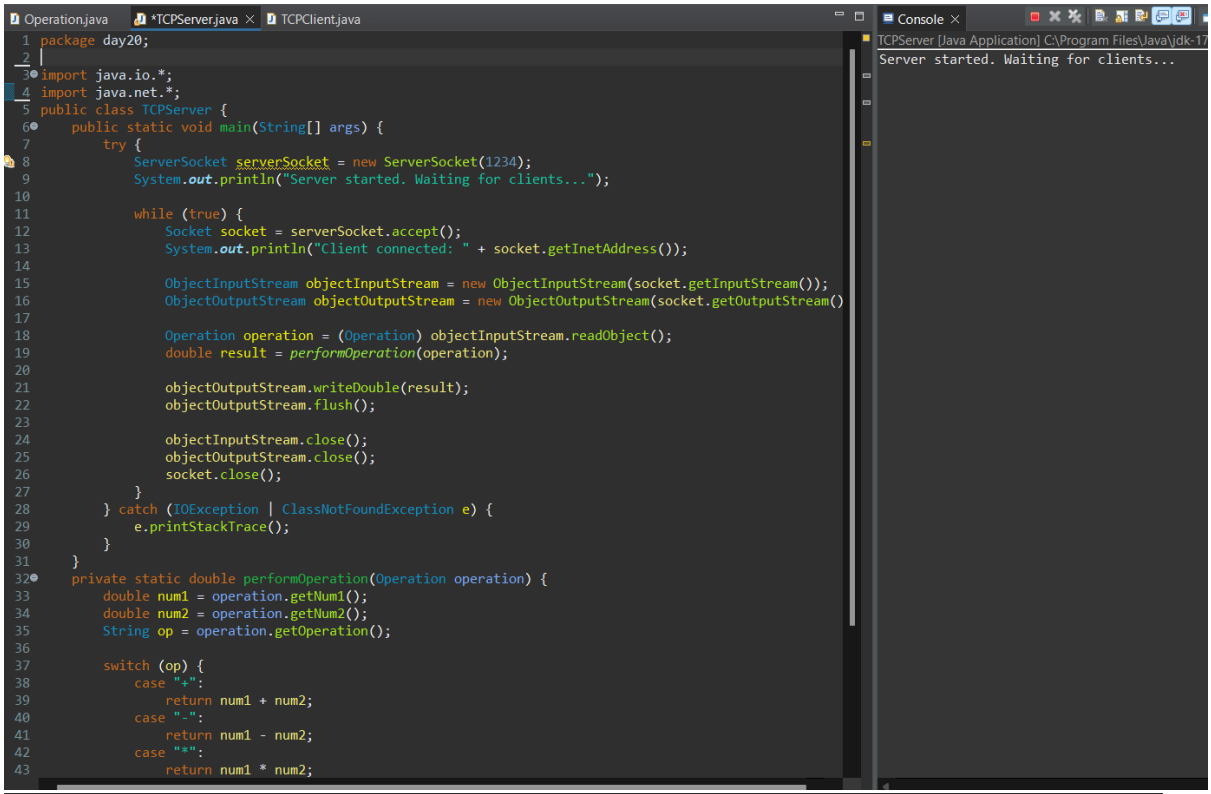
Console ×

<terminated> JavaNetworking [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (3 Jun 2024, 8:40:10 pm – 8:40:11 pm) [pid: 9608]

```
Response Code: 200
Response Headers:
null: [HTTP/1.1 200 OK]
X-Cache: [HIT]
Server: [ECAcc (nyd/D188)]
Etag: ["3147526947+ident"]
Cache-Control: [max-age=604800]
Vary: [Accept-Encoding]
Last-Modified: [Thu, 17 Oct 2019 07:18:26 GMT]
Expires: [Mon, 10 Jun 2024 15:10:11 GMT]
Content-Length: [1256]
Date: [Mon, 03 Jun 2024 15:10:11 GMT]
Age: [556819]
Content-Type: [text/html; charset=UTF-8]
Response Body:
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

Task 5: Java Networking and Serialization

Develop a basic TCP client and server application where the client sends a serialized object with 2 numbers and operation to be performed on them to the server, and the server computes the result and sends it back to the client. for eg, we could send 2, 2, "+" which would mean 2 + 2




```java
package day20;

import java.io.*;
import java.net.*;
public class TCPServer {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234);
            System.out.println("Server started. Waiting for clients...");

            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("Client connected: " + socket.getInetAddress());

                ObjectInputStream objectInputStream = new ObjectInputStream(socket.getInputStream());
                ObjectOutputStream objectOutputStream = new ObjectOutputStream(socket.getOutputStream()

                Operation operation = (Operation) objectInputStream.readObject();
                double result = performOperation(operation);

                objectOutputStream.writeDouble(result);
                objectOutputStream.flush();

                objectInputStream.close();
                objectOutputStream.close();
                socket.close();
            }
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
    private static double performOperation(Operation operation) {
        double num1 = operation.getNum1();
        double num2 = operation.getNum2();
        String op = operation.getOperation();

        switch (op) {
            case "+":
                return num1 + num2;
            case "-":
                return num1 - num2;
            case "*":
                return num1 * num2;
```

```java
        switch (op) {
            case "+":
                return num1 + num2;
            case "-":
                return num1 - num2;
            case "*":
                return num1 * num2;
            case "/":
                if (num2 != 0)
                    return num1 / num2;
                else
                    return Double.NaN;
            default:
                return Double.NaN;
        }
    }
}
```

Operation.java TCPServer.java TCPClient.java

```java
package day20;

import java.io.*;
import java.net.*;
public class TCPClient {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234);
            ObjectOutputStream objectOutputStream = new ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream objectInputStream = new ObjectInputStream(socket.getInputStream());

            Operation operation = new Operation(2, 2, "+");
            objectOutputStream.writeObject(operation);
            objectOutputStream.flush();

            double result = objectInputStream.readDouble();
            System.out.println("Result received from server: " + result);

            objectOutputStream.close();
            objectInputStream.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Console

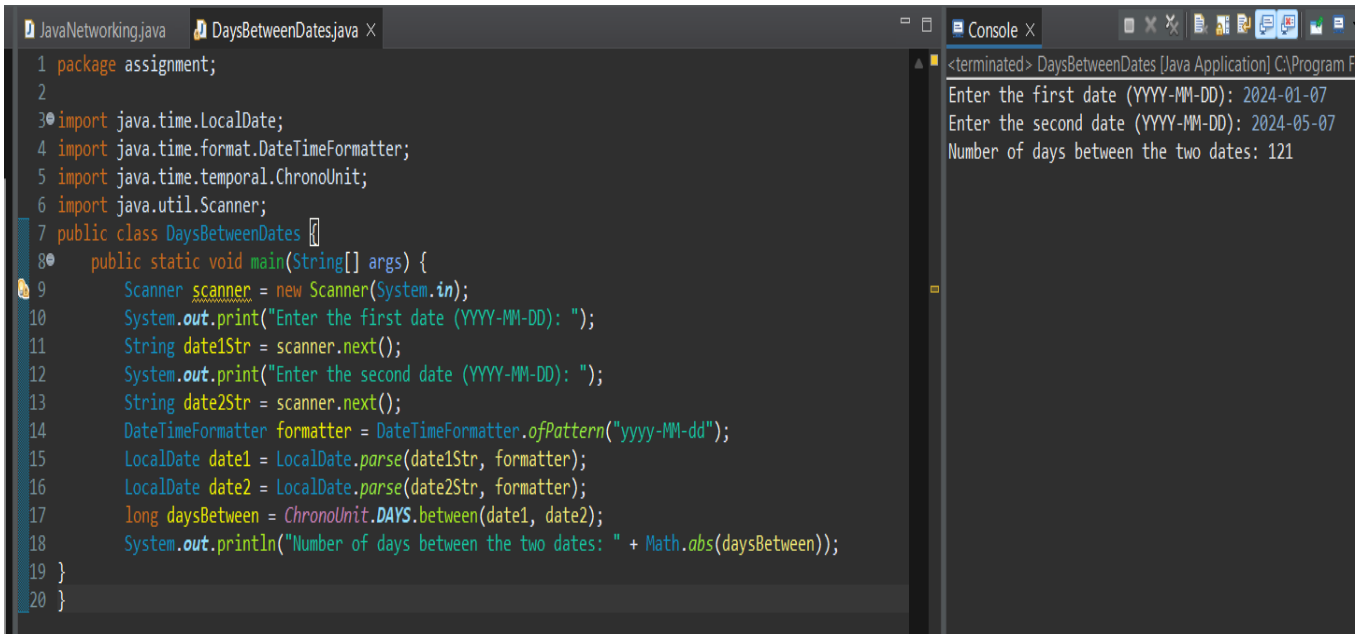
```
<terminated> TCPClient [Java Application] C:\Progra
Result received from server: 4.0
```

Operation.java TCPServer.java TCPClient.java

```java
package day20;

import java.io.Serializable;

public class Operation implements Serializable {
    private static final long serialVersionUID = 1L;
    private double num1;
    private double num2;
    private String operation;
    public Operation(double num1, double num2, String operation) {
        this.num1 = num1;
        this.num2 = num2;
        this.operation = operation;
    }
    public double getNum1() {
        return num1;
    }
    public double getNum2() {
        return num2;
    }
    public String getOperation() {
        return operation;
    }
}
```

Task 6: Java 8 Date and Time API

Write a program that calculates the number of days between two dates input by the user.



Task 7: Timezone Create a timezone converter that takes a time in one timezone and converts it to another timezone.

TimezoneConverter.java

```java
package assignment;

import java.time.*;

public class TimezoneConverter {
    public static void main(String[] args) {
        LocalDateTime dateTime = LocalDateTime.of(2024, 6, 3, 12, 0);
        ZoneId fromZone = ZoneId.of("America/New_York");
        ZoneId toZone = ZoneId.of("Asia/Tokyo");
        ZonedDateTime fromZonedDateTime = ZonedDateTime.of(dateTime, fromZone);
        ZonedDateTime toZonedDateTime = fromZonedDateTime.withZoneSameInstant(toZone);
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
        String fromTime = fromZonedDateTime.format(formatter);
        String toTime = toZonedDateTime.format(formatter);
        System.out.println("Time in " + fromZone + ": " + fromTime);
        System.out.println("Converted time in " + toZone + ": " + toTime);
    }
}
```

Console

```
<terminated> TimezoneConverter [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (3 Jun 2024, 8:46:15 pm – 8:46:15 pm) [pid: 8244]
Time in America/New_York: 2024-06-03 12:00:00
Converted time in Asia/Tokyo: 2024-06-04 01:00:00
```