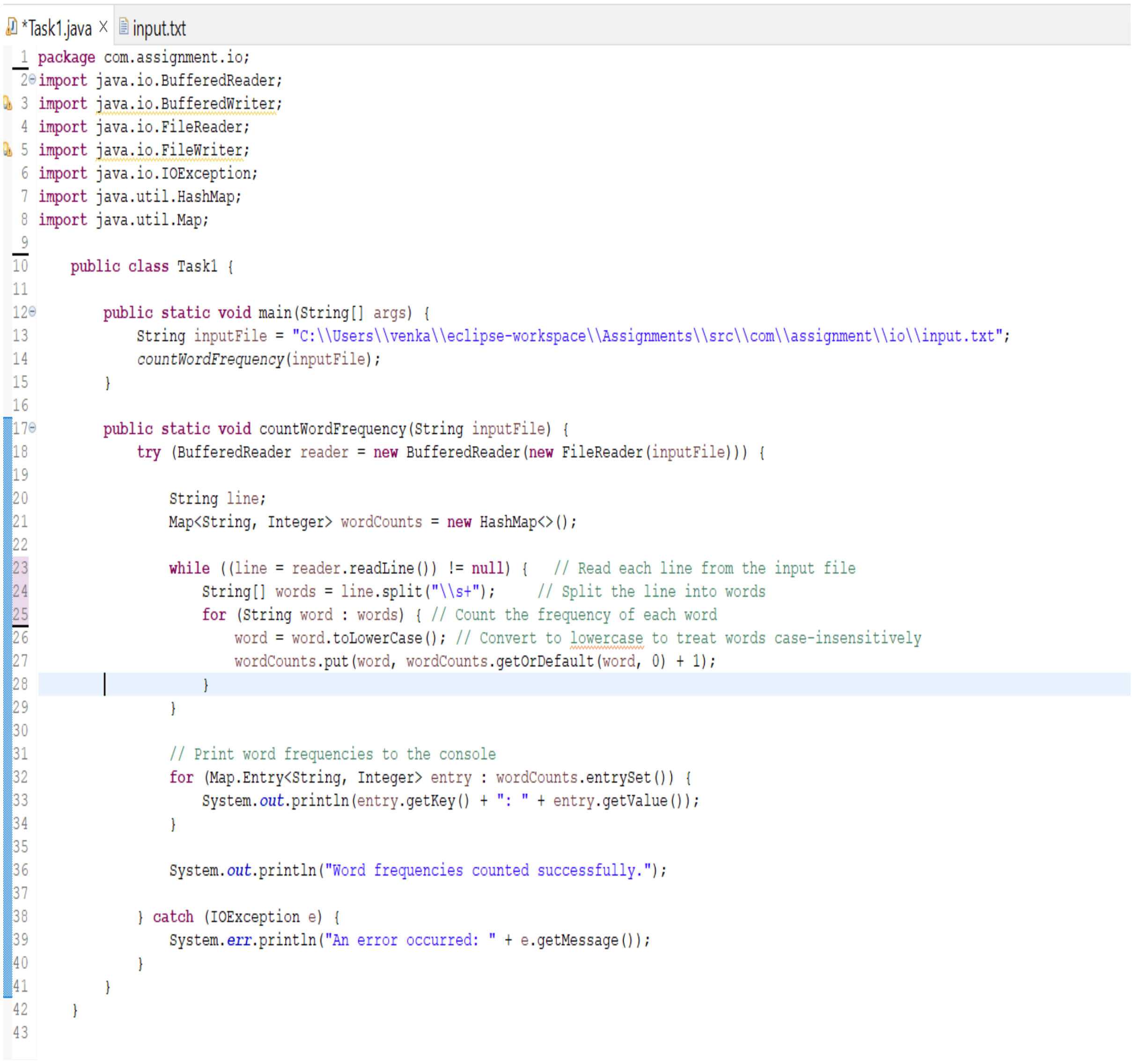
Day 20:

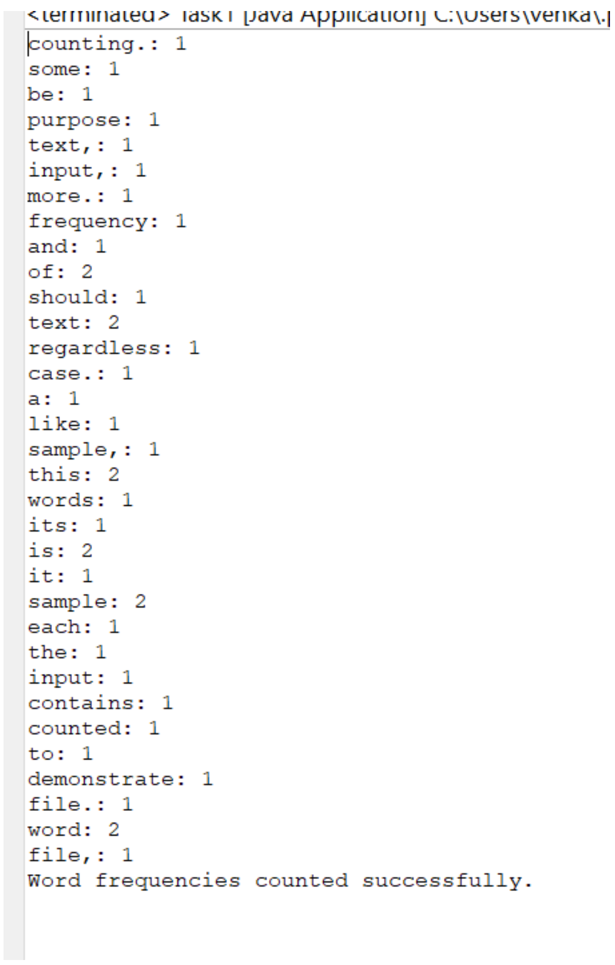
Task 1: Java IO Basics

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



```java
package com.assignment.io;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

    public class Task1 {

        public static void main(String[] args) {
            String inputFile = "C:\\Users\\venka\\eclipse-workspace\\Assignments\\src\\com\\assignment\\io\\input.txt";
            countWordFrequency(inputFile);
        }

        public static void countWordFrequency(String inputFile) {
            try (BufferedReader reader = new BufferedReader(new FileReader(inputFile))) {

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

                while ((line = reader.readLine()) != null) {   // Read each line from the input file
                    String[] words = line.split("\\s+");     // Split the line into words
                    for (String word : words) { // Count the frequency of each word
                        word = word.toLowerCase(); // Convert to lowercase to treat words case-insensitively
                        wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);
                    }
                }

                // Print word frequencies to the console
                for (Map.Entry<String, Integer> entry : wordCounts.entrySet()) {
                    System.out.println(entry.getKey() + ": " + entry.getValue());
                }

                System.out.println("Word frequencies counted successfully.");

            } catch (IOException e) {
                System.err.println("An error occurred: " + e.getMessage());
            }
        }
    }
```

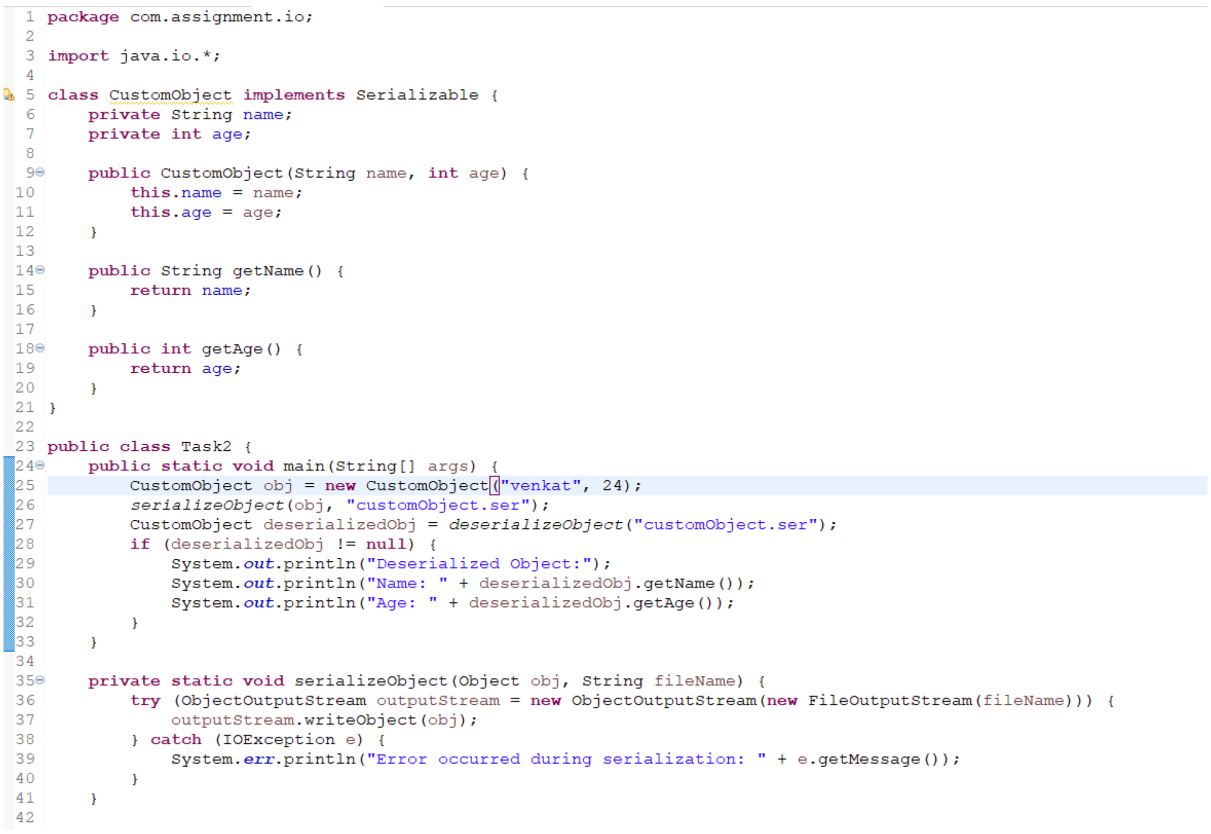
OUTPUT:



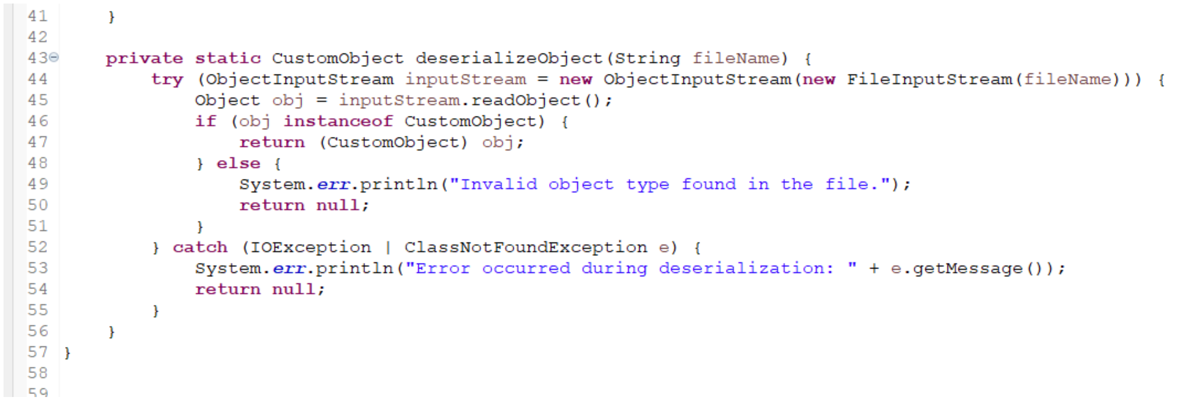
```
<terminated> Task1 [Java Application] C:\Users\venka\.
counting.: 1
some: 1
be: 1
purpose: 1
text,: 1
input,: 1
more.: 1
frequency: 1
and: 1
of: 2
should: 1
text: 2
regardless: 1
case.: 1
a: 1
like: 1
sample,: 1
this: 2
words: 1
its: 1
is: 2
it: 1
sample: 2
each: 1
the: 1
input: 1
contains: 1
counted: 1
to: 1
demonstrate: 1
file.: 1
word: 2
file,: 1
Word frequencies counted successfully.
```

Task 2: Serialization and Deserialization

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

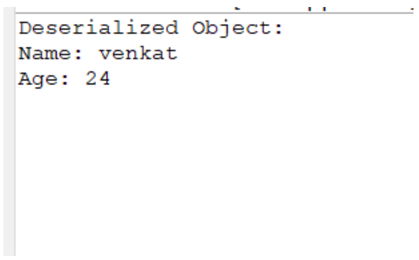


```java
package com.assignment.io;

import java.io.*;

class CustomObject implements Serializable {
    private String name;
    private int age;

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

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}

public class Task2 {
    public static void main(String[] args) {
        CustomObject obj = new CustomObject("venkat", 24);
        serializeObject(obj, "customObject.ser");
        CustomObject deserializedObj = deserializeObject("customObject.ser");
        if (deserializedObj != null) {
            System.out.println("Deserialized Object:");
            System.out.println("Name: " + deserializedObj.getName());
            System.out.println("Age: " + deserializedObj.getAge());
        }
    }

    private static void serializeObject(Object obj, String fileName) {
        try (ObjectOutputStream outputStream = new ObjectOutputStream(new FileOutputStream(fileName))) {
            outputStream.writeObject(obj);
        } catch (IOException e) {
            System.err.println("Error occurred during serialization: " + e.getMessage());
        }
    }

```



```java
    }

    private static CustomObject deserializeObject(String fileName) {
        try (ObjectInputStream inputStream = new ObjectInputStream(new FileInputStream(fileName))) {
            Object obj = inputStream.readObject();
            if (obj instanceof CustomObject) {
                return (CustomObject) obj;
            } else {
                System.err.println("Invalid object type found in the file.");
                return null;
            }
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error occurred during deserialization: " + e.getMessage());
            return null;
        }
    }
}

```

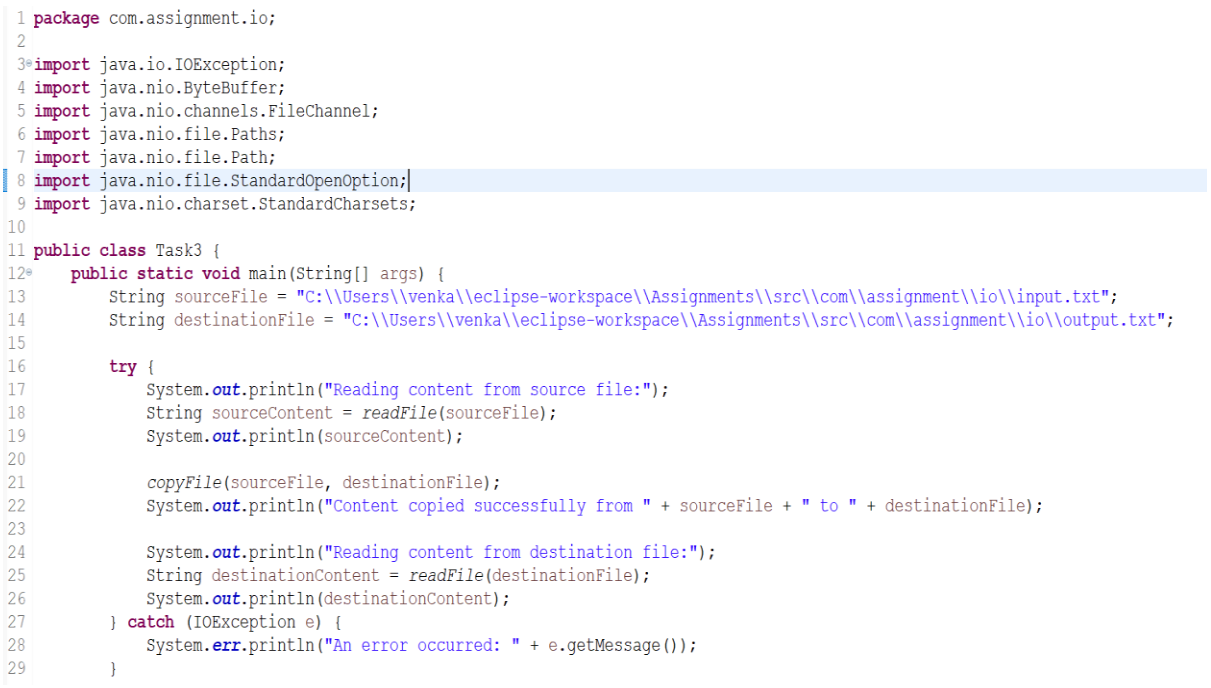
OUTPUT ;



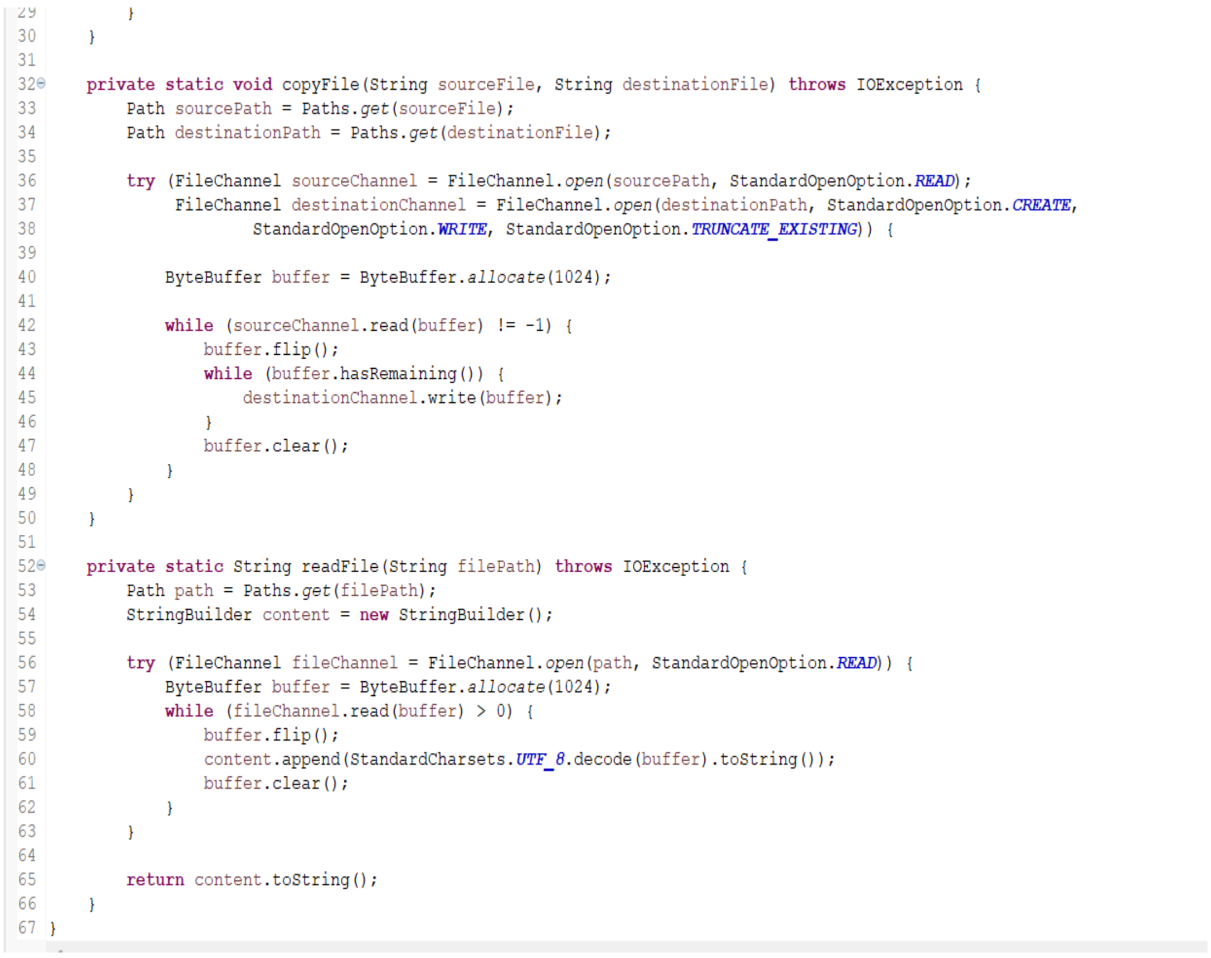
```
Deserialized Object:
Name: venkat
Age: 24
```

Task 3: New IO (NIO)

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

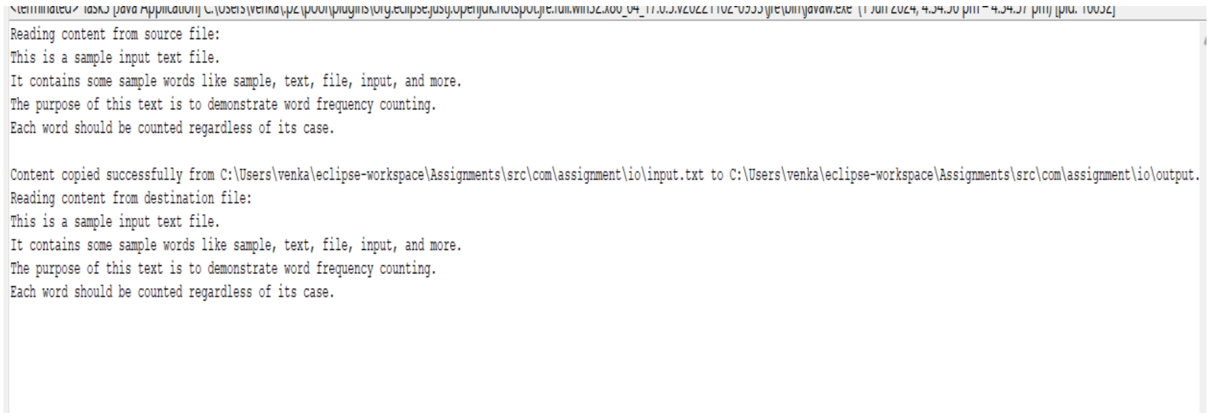


```java
package com.assignment.io;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Paths;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.nio.charset.StandardCharsets;

public class Task3 {
    public static void main(String[] args) {
        String sourceFile = "C:\\Users\\venka\\eclipse-workspace\\Assignments\\src\\com\\assignment\\io\\input.txt";
        String destinationFile = "C:\\Users\\venka\\eclipse-workspace\\Assignments\\src\\com\\assignment\\io\\output.txt";

        try {
            System.out.println("Reading content from source file:");
            String sourceContent = readFile(sourceFile);
            System.out.println(sourceContent);

            copyFile(sourceFile, destinationFile);
            System.out.println("Content copied successfully from " + sourceFile + " to " + destinationFile);

            System.out.println("Reading content from destination file:");
            String destinationContent = readFile(destinationFile);
            System.out.println(destinationContent);
        } catch (IOException e) {
            System.err.println("An error occurred: " + e.getMessage());
        }
```



```java
        }
    }

    private static void copyFile(String sourceFile, String destinationFile) throws IOException {
        Path sourcePath = Paths.get(sourceFile);
        Path destinationPath = Paths.get(destinationFile);

        try (FileChannel sourceChannel = FileChannel.open(sourcePath, StandardOpenOption.READ);
             FileChannel destinationChannel = FileChannel.open(destinationPath, StandardOpenOption.CREATE,
                     StandardOpenOption.WRITE, StandardOpenOption.TRUNCATE_EXISTING)) {

            ByteBuffer buffer = ByteBuffer.allocate(1024);

            while (sourceChannel.read(buffer) != -1) {
                buffer.flip();
                while (buffer.hasRemaining()) {
                    destinationChannel.write(buffer);
                }
                buffer.clear();
            }
        }
    }

    private static String readFile(String filePath) throws IOException {
        Path path = Paths.get(filePath);
        StringBuilder content = new StringBuilder();

        try (FileChannel fileChannel = FileChannel.open(path, StandardOpenOption.READ)) {
            ByteBuffer buffer = ByteBuffer.allocate(1024);
            while (fileChannel.read(buffer) > 0) {
                buffer.flip();
                content.append(StandardCharsets.UTF_8.decode(buffer).toString());
                buffer.clear();
            }
        }

        return content.toString();
    }
}
```

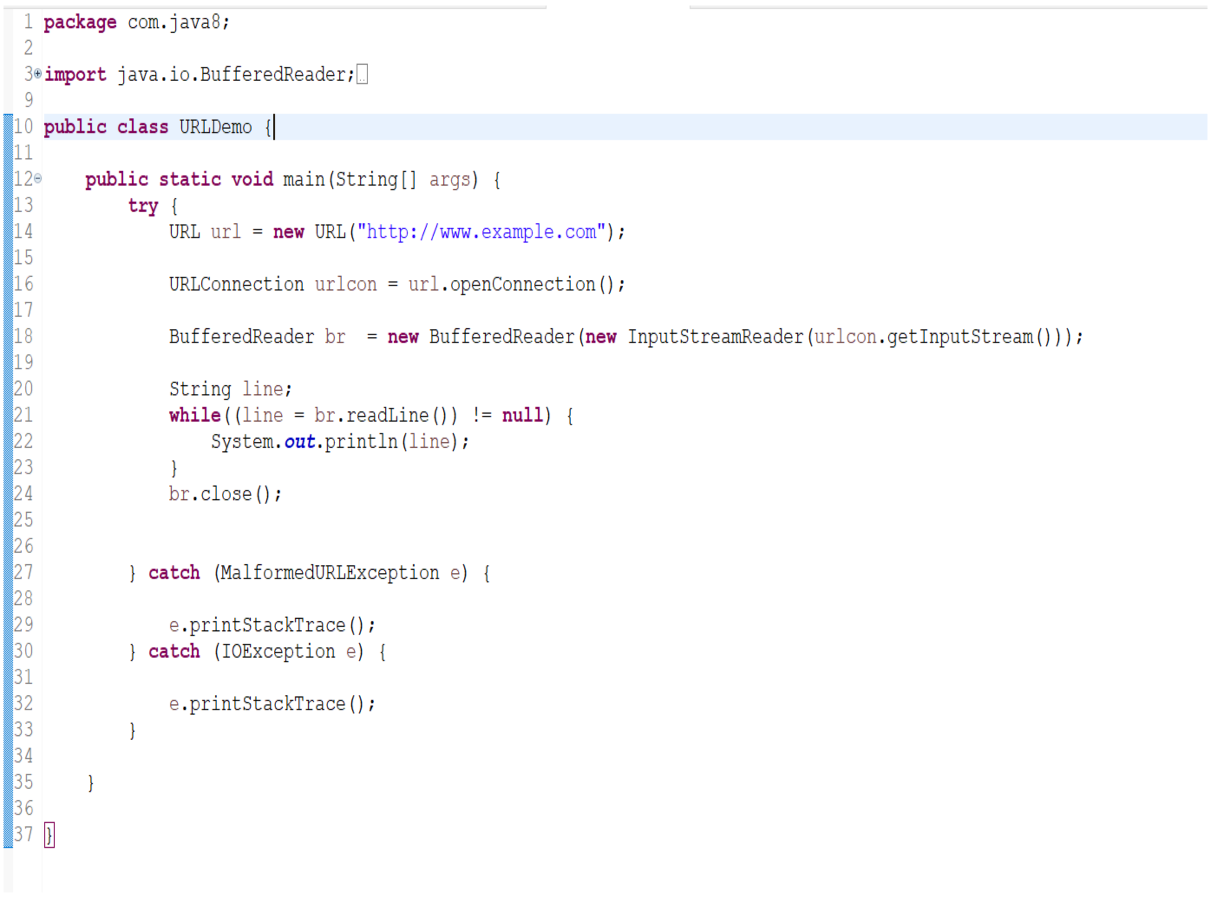
Output:



```
<terminated> Task3 [Java Application] C:\Users\venka\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (1 Jun 2024, 4:34:36 pm – 4:34:37 pm) [pid: 10052]
Reading content from source file:
This is a sample input text file.
It contains some sample words like sample, text, file, input, and more.
The purpose of this text is to demonstrate word frequency counting.
Each word should be counted regardless of its case.

Content copied successfully from C:\Users\venka\eclipse-workspace\Assignments\src\com\assignment\io\input.txt to C:\Users\venka\eclipse-workspace\Assignments\src\com\assignment\io\output.
Reading content from destination file:
This is a sample input text file.
It contains some sample words like sample, text, file, input, and more.
The purpose of this text is to demonstrate word frequency counting.
Each word should be counted regardless of its case.
```

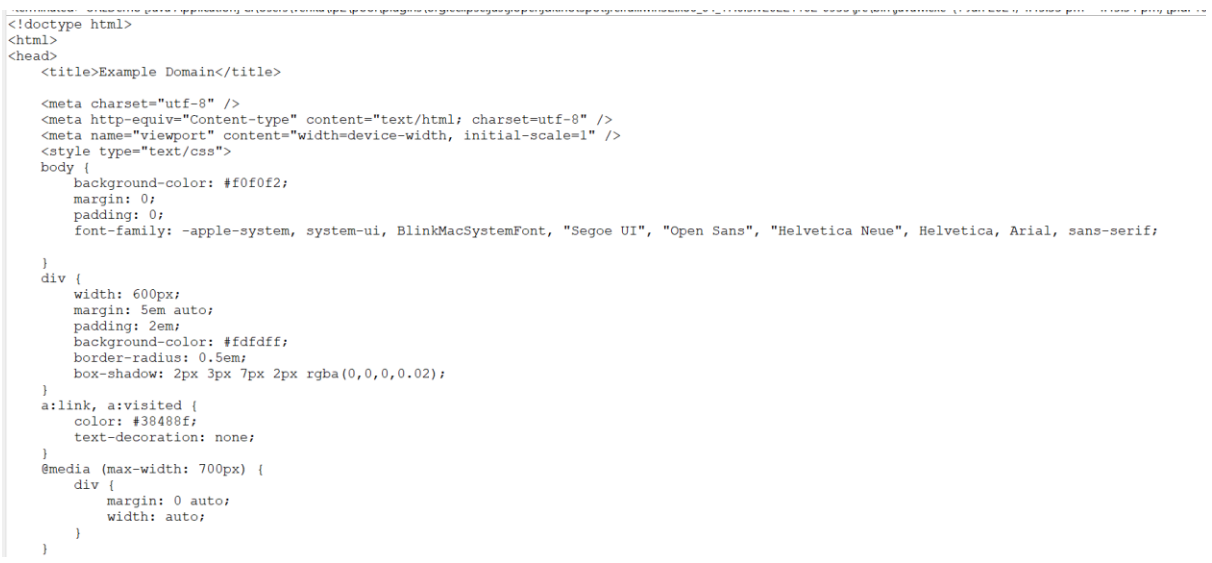
Task 4: Java Networking

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

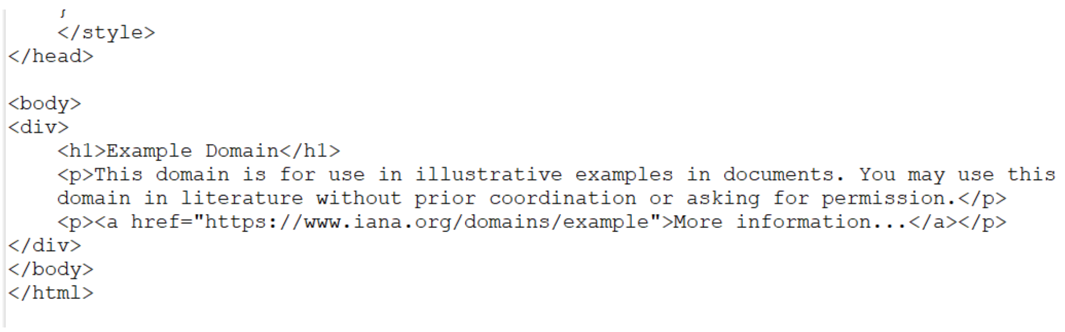


```java
package com.java8;

import java.io.BufferedReader;

public class URLDemo {

    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.example.com");

            URLConnection urlcon = url.openConnection();

            BufferedReader br  = new BufferedReader(new InputStreamReader(urlcon.getInputStream()));

            String line;
            while((line = br.readLine()) != null) {
                System.out.println(line);
            }
            br.close();


        } catch (MalformedURLException e) {

            e.printStackTrace();
        } catch (IOException e) {

            e.printStackTrace();
        }

    }

}
```

OUTPUT:



```
<!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;
        }
    }
```

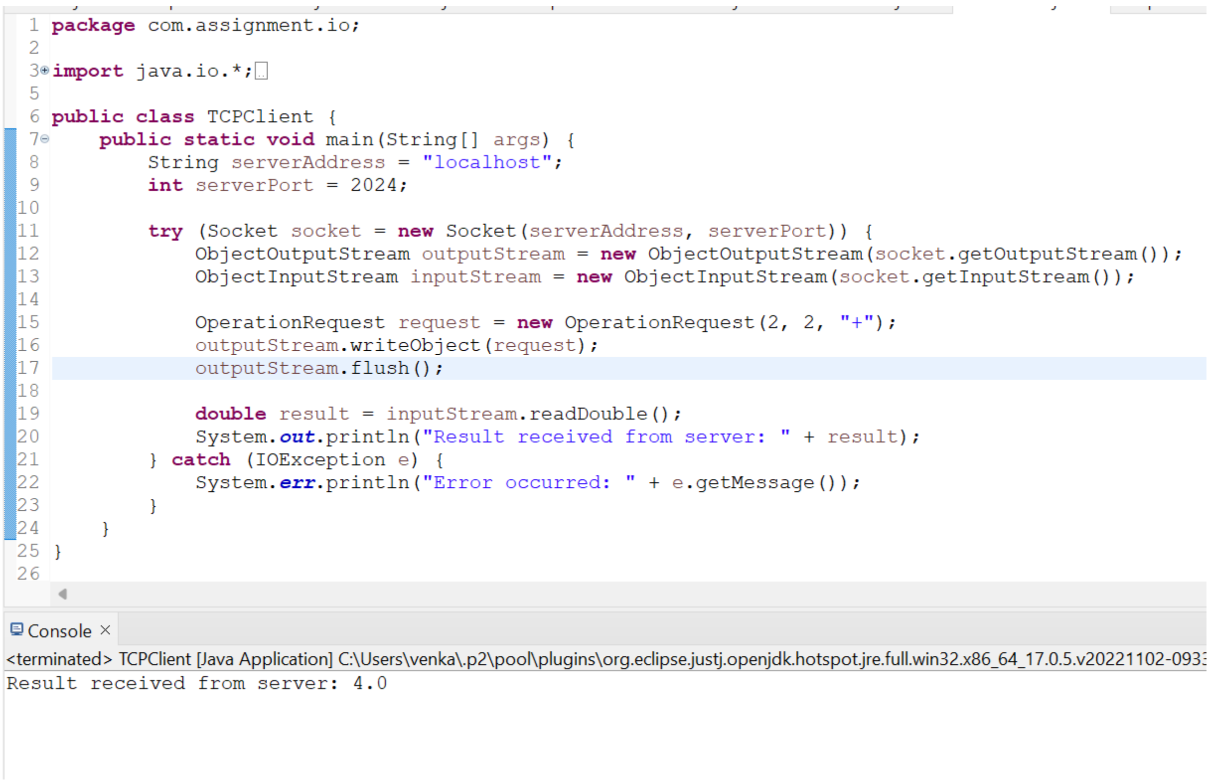


```
    }
    </style>
</head>

<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.

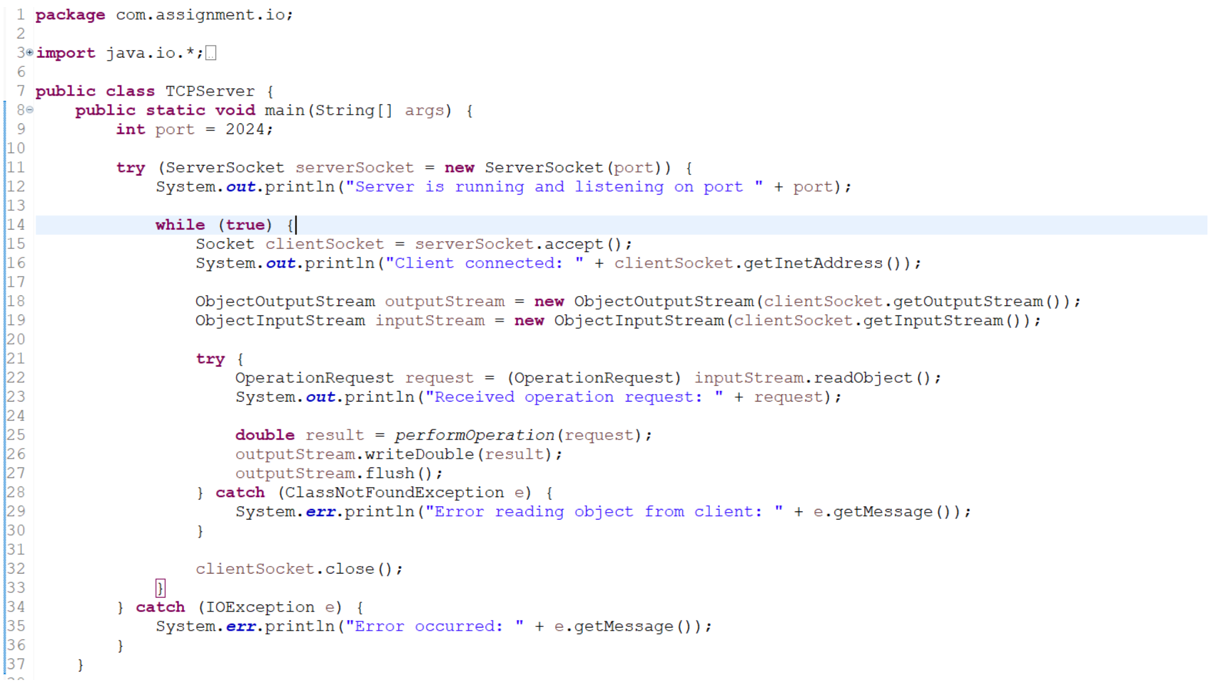
CLIENT:



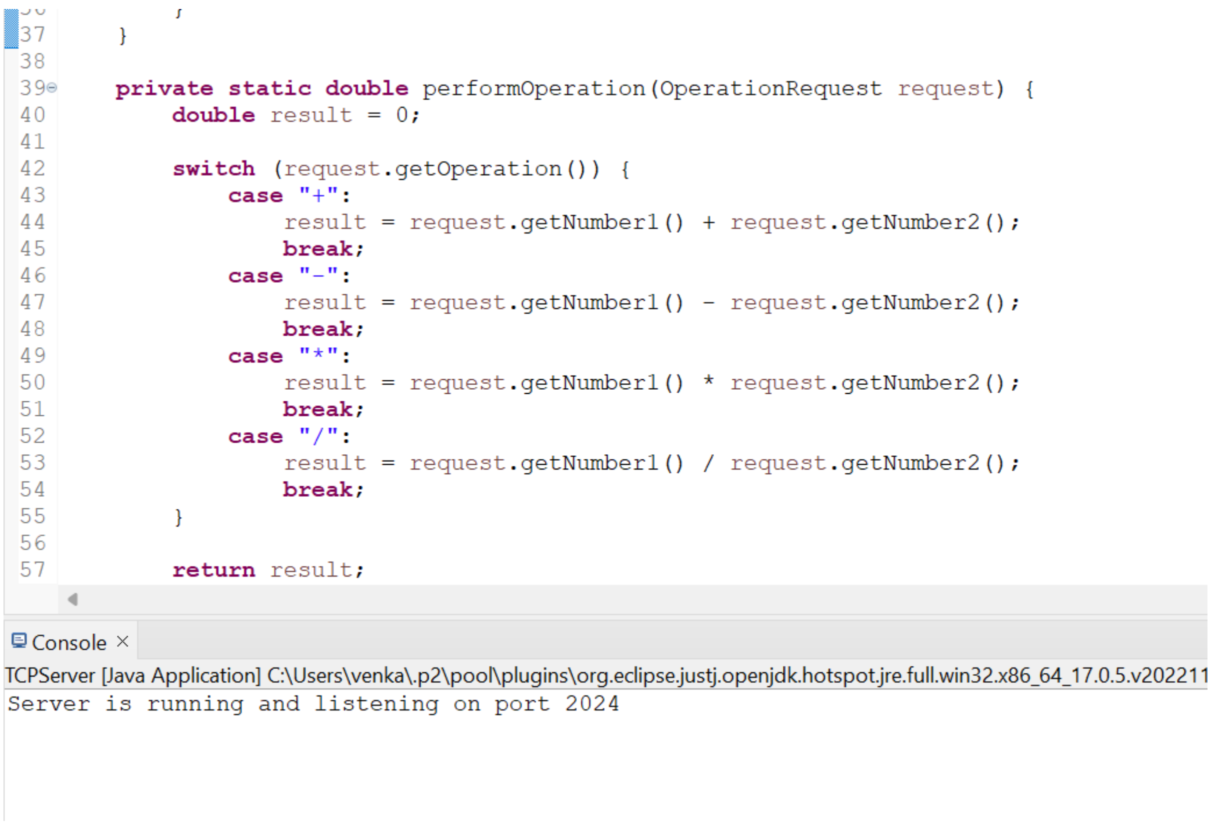
```
package com.assignment.io;

import java.io.*;

public class TCPClient {
    public static void main(String[] args) {
        String serverAddress = "localhost";
        int serverPort = 2024;

        try (Socket socket = new Socket(serverAddress, serverPort)) {
            ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

            OperationRequest request = new OperationRequest(2, 2, "+");
            outputStream.writeObject(request);
            outputStream.flush();

            double result = inputStream.readDouble();
            System.out.println("Result received from server: " + result);
        } catch (IOException e) {
            System.err.println("Error occurred: " + e.getMessage());
        }
    }
}

```

```
<terminated> TCPClient [Java Application] C:\Users\venka\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-093
Result received from server: 4.0
```

SERVER :



```java
package com.assignment.io;

import java.io.*;

public class TCPServer {
    public static void main(String[] args) {
        int port = 2024;

        try (ServerSocket serverSocket = new ServerSocket(port)) {
            System.out.println("Server is running and listening on port " + port);

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

                ObjectOutputStream outputStream = new ObjectOutputStream(clientSocket.getOutputStream());
                ObjectInputStream inputStream = new ObjectInputStream(clientSocket.getInputStream());

                try {
                    OperationRequest request = (OperationRequest) inputStream.readObject();
                    System.out.println("Received operation request: " + request);

                    double result = performOperation(request);
                    outputStream.writeDouble(result);
                    outputStream.flush();
                } catch (ClassNotFoundException e) {
                    System.err.println("Error reading object from client: " + e.getMessage());
                }

                clientSocket.close();
            }
        } catch (IOException e) {
            System.err.println("Error occurred: " + e.getMessage());
        }
    }
```



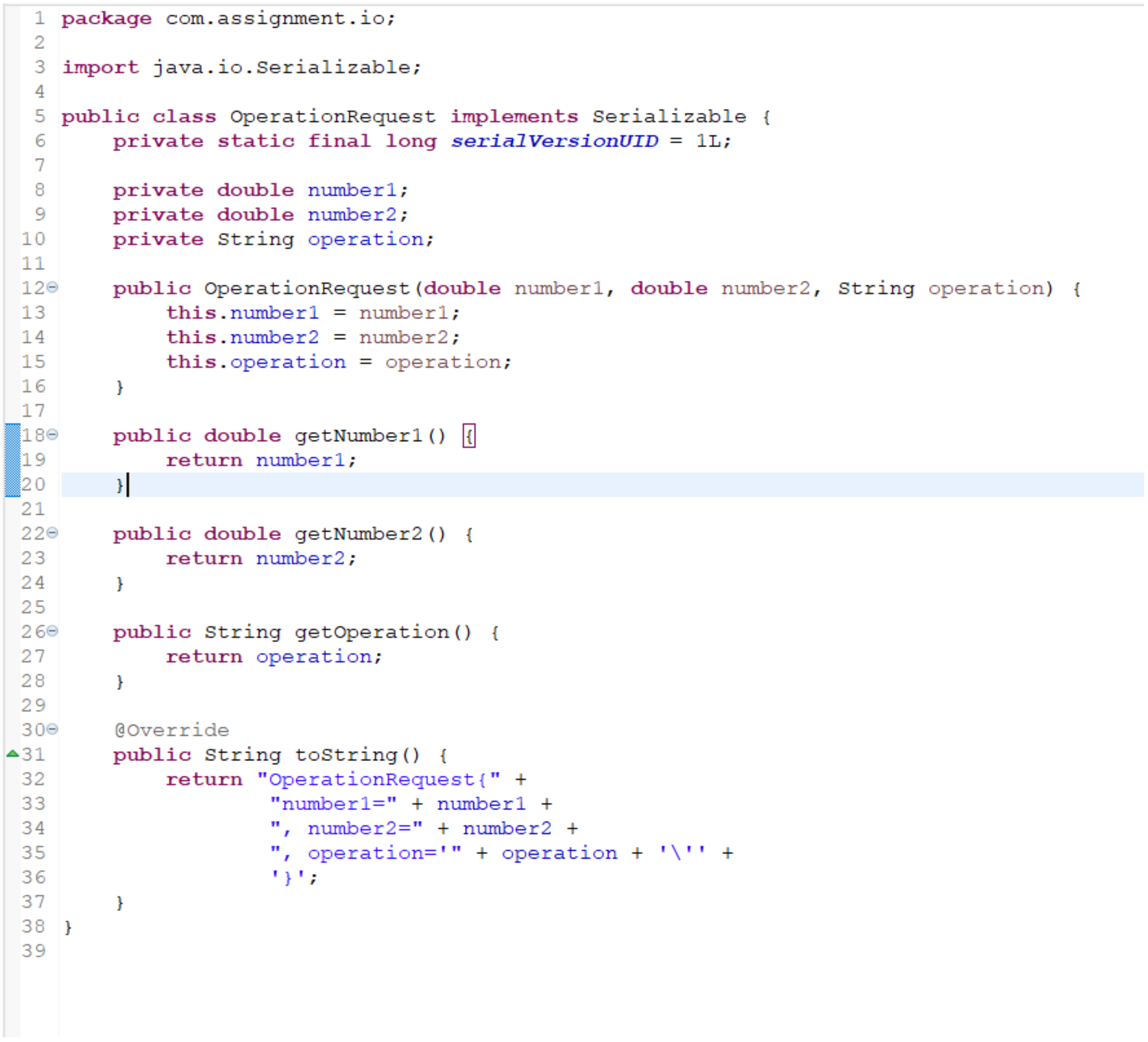
```java
    }

    private static double performOperation(OperationRequest request) {
        double result = 0;

        switch (request.getOperation()) {
            case "+":
                result = request.getNumber1() + request.getNumber2();
                break;
            case "-":
                result = request.getNumber1() - request.getNumber2();
                break;
            case "*":
                result = request.getNumber1() * request.getNumber2();
                break;
            case "/":
                result = request.getNumber1() / request.getNumber2();
                break;
        }

        return result;
```

Console

TCPServer [Java Application] C:\Users\venka\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v202211

```
Server is running and listening on port 2024
```

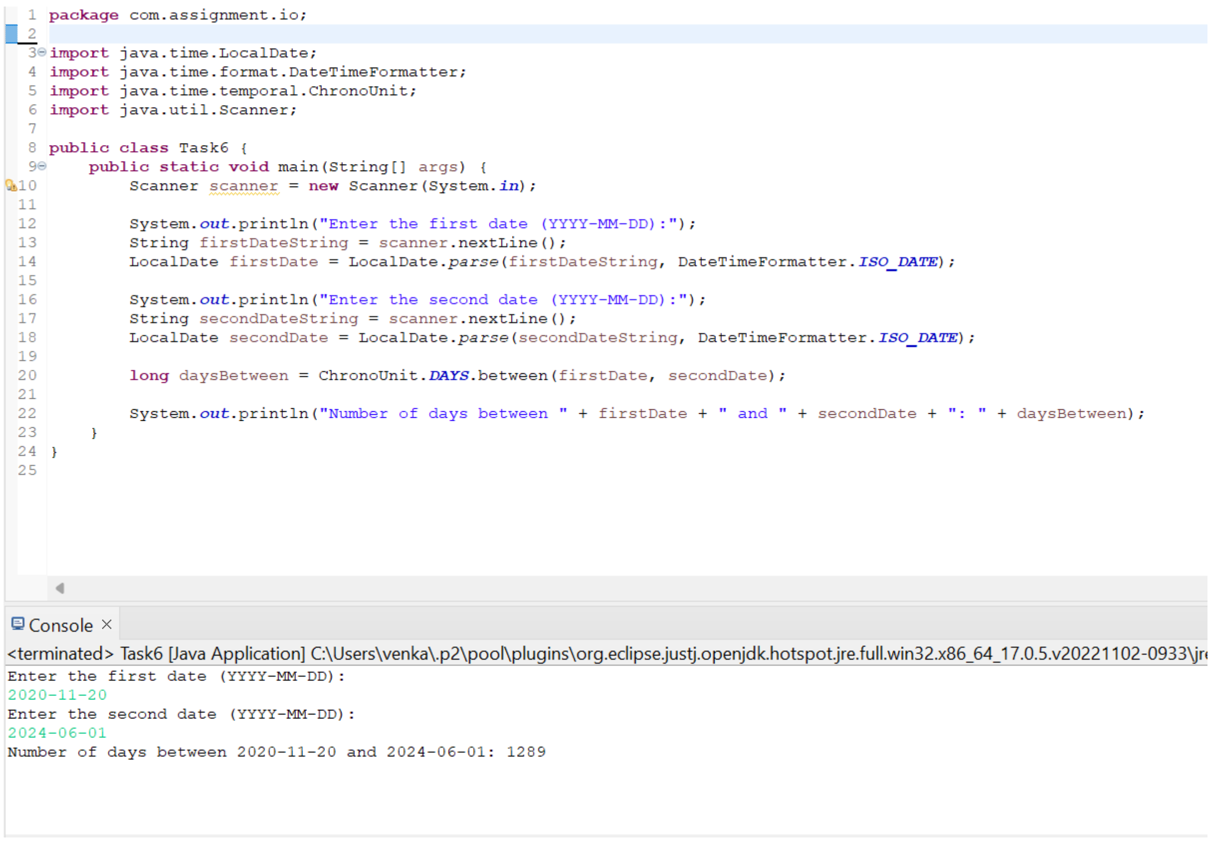
Operation Request:



```java
package com.assignment.io;

import java.io.Serializable;

public class OperationRequest implements Serializable {
    private static final long serialVersionUID = 1L;

    private double number1;
    private double number2;
    private String operation;

    public OperationRequest(double number1, double number2, String operation) {
        this.number1 = number1;
        this.number2 = number2;
        this.operation = operation;
    }

    public double getNumber1() {
        return number1;
    }

    public double getNumber2() {
        return number2;
    }

    public String getOperation() {
        return operation;
    }

    @Override
    public String toString() {
        return "OperationRequest{" +
                "number1=" + number1 +
                ", number2=" + number2 +
                ", operation='" + 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.

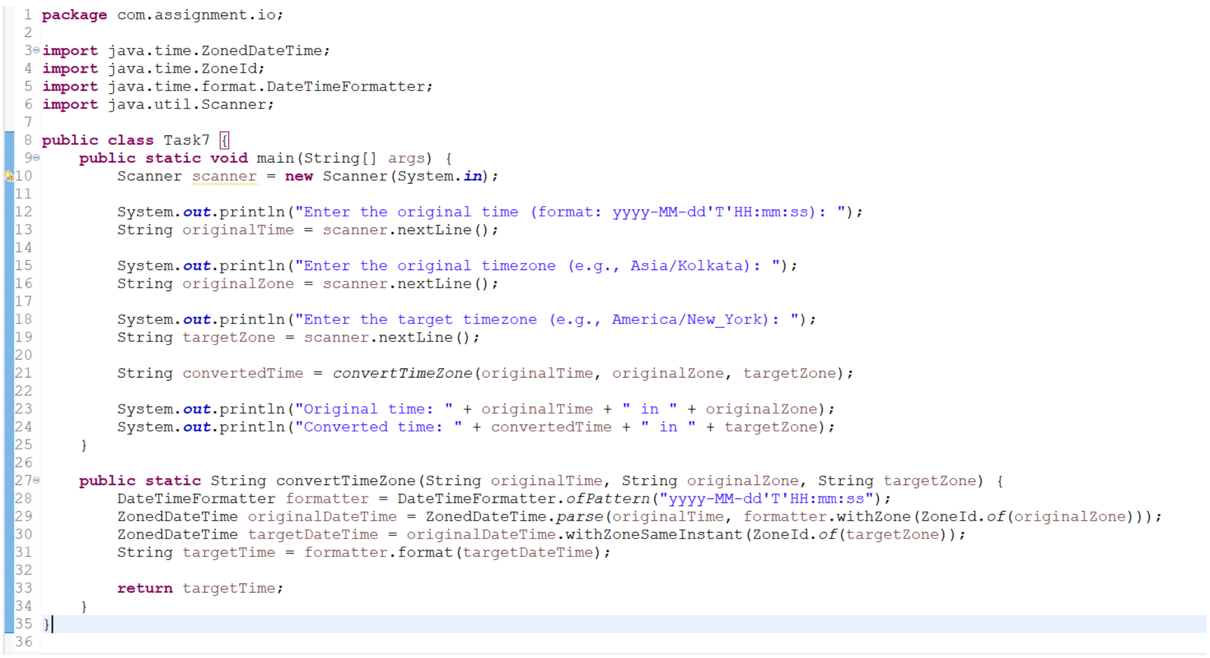


```java
package com.assignment.io;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;

public class Task6 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the first date (YYYY-MM-DD):");
        String firstDateString = scanner.nextLine();
        LocalDate firstDate = LocalDate.parse(firstDateString, DateTimeFormatter.ISO_DATE);

        System.out.println("Enter the second date (YYYY-MM-DD):");
        String secondDateString = scanner.nextLine();
        LocalDate secondDate = LocalDate.parse(secondDateString, DateTimeFormatter.ISO_DATE);

        long daysBetween = ChronoUnit.DAYS.between(firstDate, secondDate);

        System.out.println("Number of days between " + firstDate + " and " + secondDate + ": " + daysBetween);
    }
}

```

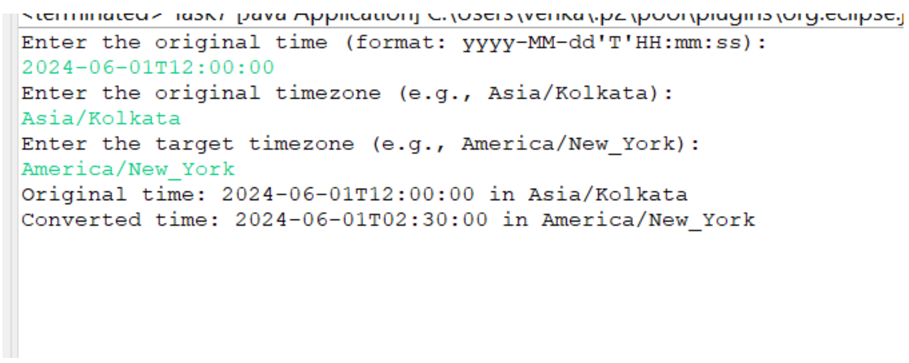
```
<terminated> Task6 [Java Application] C:\Users\venka\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jr
Enter the first date (YYYY-MM-DD):
2020-11-20
Enter the second date (YYYY-MM-DD):
2024-06-01
Number of days between 2020-11-20 and 2024-06-01: 1289
```

Task 7: Timezone

Create a timezone converter that takes a time in one timezone and converts it to another timezone.



```java
package com.assignment.io;

import java.time.ZonedDateTime;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class Task7 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the original time (format: yyyy-MM-dd'T'HH:mm:ss): ");
        String originalTime = scanner.nextLine();

        System.out.println("Enter the original timezone (e.g., Asia/Kolkata): ");
        String originalZone = scanner.nextLine();

        System.out.println("Enter the target timezone (e.g., America/New_York): ");
        String targetZone = scanner.nextLine();

        String convertedTime = convertTimeZone(originalTime, originalZone, targetZone);

        System.out.println("Original time: " + originalTime + " in " + originalZone);
        System.out.println("Converted time: " + convertedTime + " in " + targetZone);
    }

    public static String convertTimeZone(String originalTime, String originalZone, String targetZone) {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd'T'HH:mm:ss");
        ZonedDateTime originalDateTime = ZonedDateTime.parse(originalTime, formatter.withZone(ZoneId.of(originalZone)));
        ZonedDateTime targetDateTime = originalDateTime.withZoneSameInstant(ZoneId.of(targetZone));
        String targetTime = formatter.format(targetDateTime);

        return targetTime;
    }
}
```



```
<terminated> Task7 [Java Application] C:\Users\venka\.p2\pool\plugins\org.eclipse.
Enter the original time (format: yyyy-MM-dd'T'HH:mm:ss):
2024-06-01T12:00:00
Enter the original timezone (e.g., Asia/Kolkata):
Asia/Kolkata
Enter the target timezone (e.g., America/New_York):
America/New_York
Original time: 2024-06-01T12:00:00 in Asia/Kolkata
Converted time: 2024-06-01T02:30:00 in America/New_York
```