

Assignments

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`.

```
1 package assignment;
2
3 import java.io.*;
4 import java.util.HashMap;
5 import java.util.Map;
6 import java.util.Scanner;
7
8 public class WordFrequencyCounter {
9     public static void main(String[] args) {
10         if (args.length != 2) {
11             System.out.println("Usage: java WordFrequencyCounter <inputFile> <outputFile>");
12             return;
13         }
14
15         String inputFileName = args[0];
16         String outputFileName = args[1];
17
18         Map<String, Integer> wordCounts = new HashMap<>();
19
20         try (FileReader fileReader = new FileReader(inputFileName);
21             Scanner scanner = new Scanner(fileReader)) {
22
23             while (scanner.hasNext()) {
24                 String word = scanner.next().toLowerCase().replaceAll("[^a-zA-Z]", "");
25                 if (!word.isEmpty()) {
26                     wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);
27                 }
28             }
29         } catch (IOException e) {
30             System.out.println("An error occurred while reading the file: " + e.getMessage());
31         }
32
33         try (FileWriter fileWriter = new FileWriter(outputFileName)) {
34             for (Map.Entry<String, Integer> entry : wordCounts.entrySet()) {
35                 fileWriter.write(entry.getKey() + ": " + entry.getValue() + System.lineSeparator());
36             }
37         } catch (IOException e) {
38             System.out.println("An error occurred while writing to the file: " + e.getMessage());
39         }
40
41         System.out.println("Word frequency count has been written to " + outputFileName);
42     }
43 }
```

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

WordFrequencyCounter.java input.txt output.txt

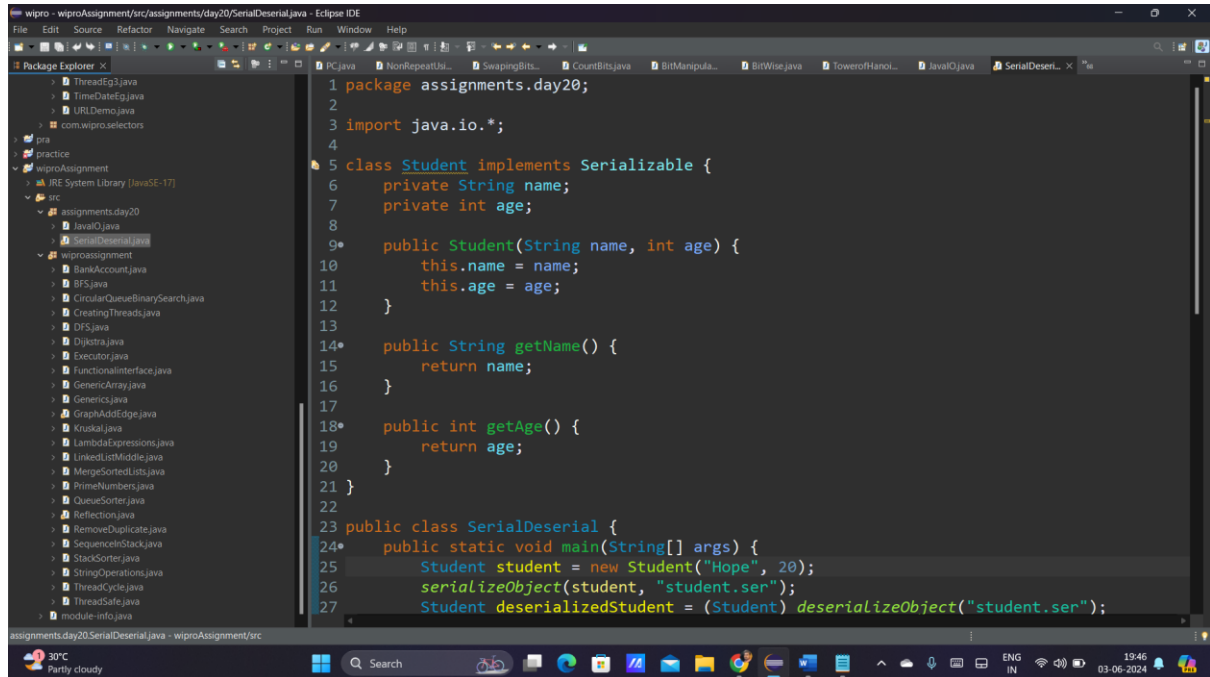
1Hi My name is Divya.

WordFrequencyCounter.java input.txt output.txt X

1hi: 1
2name: 1
3is: 1
4divya: 1
5my: 1
6

Task 2: Serialization and Deserialization

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



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project named 'wiproAssignment' with a package 'assignments.day20'. The main editor window shows the following Java code:

```
1 package assignments.day20;
2
3 import java.io.*;
4
5 class Student implements Serializable {
6     private String name;
7     private int age;
8
9     public Student(String name, int age) {
10         this.name = name;
11         this.age = age;
12     }
13
14     public String getName() {
15         return name;
16     }
17
18     public int getAge() {
19         return age;
20     }
21 }
22
23 public class SerialDeserial {
24     public static void main(String[] args) {
25         Student student = new Student("Hope", 20);
26         serializeObject(student, "student.ser");
27         Student deserializedStudent = (Student) deserializeObject("student.ser");
```

The bottom status bar shows the system temperature as 30°C, the time as 19:46, and the date as 03-06-2024.

```
23 public class SerialDeserial {
24*   public static void main(String[] args) {
25       Student student = new Student("Hope", 20);
26       serializeObject(student, "student.ser");
27       Student deserializedStudent = (Student) deserializeObject("student.ser");
28       System.out.println("Serialized Student:");
29       System.out.println("Name: " + deserializedStudent.getName());
30       System.out.println("Age: " + deserializedStudent.getAge());
31   }
32
33*   public static void serializeObject(Object obj, String fileName) {
34       try {
35           FileOutputStream fileOut = new FileOutputStream(fileName);
36           ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);
37           objectOut.writeObject(obj);
38           objectOut.close();
39           fileOut.close();
40           System.out.println("Object serialized successfully.");
41       } catch (IOException e) {
42           System.err.println("Error during serialization: " + e.getMessage());
43       }
44   }
45
46*   public static Object deserializeObject(String fileName) {
47       try {
48           FileInputStream fileIn = new FileInputStream(fileName);
49           ObjectInputStream objectIn = new ObjectInputStream(fileIn);
50           Object obj = objectIn.readObject();
51           objectIn.close();
52           fileIn.close();
53           System.out.println("Object deserialized successfully.");
54           return obj;
55       } catch (IOException | ClassNotFoundException e) {
56           System.err.println("Error during deserialization: " + e.getMessage());
57           return null;
58       }
59   }
60 }
61
```

Console Output:

```
Object serialized successfully.
Object deserialized successfully.
Deserialized Student:
Name: Hope
Age: 20
```

Task 3: New IO (NIO)

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

The screenshot shows the Eclipse IDE with the file `MnioC.java` open. The Package Explorer on the left shows a project named `com.wipro` with various sub-packages and classes. The main editor displays the following Java code:

```
1 package com.wipro;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12 public class MnioC {
13     String fileName = "mydir/rhymes.txt";
14
15     public void createDirectory() {
16         Path p = Paths.get("mydir");
17         try {
18             if (Files.exists(p)) {
19                 System.out.println("Directory already exists");
20             } else {
21                 Path cPath = Files.createDirectories(p);
22                 System.out.println("Directory created at " + cPath.toString());
23             }
24         } catch (Exception e) {
25             e.printStackTrace();
26         }
27     }
28
29     public void createFile(String fileName) {
30         Path f = Paths.get(fileName);
31         try {
32             if (Files.exists(f)) {
33                 System.out.println("File already exists");
34             }
35         }
36     }
37 }
```

The screenshot shows the Eclipse IDE with the file `MnioC.java` open. The Package Explorer on the left shows the same project structure as the previous screenshot. The main editor displays the following Java code:

```
28
29 public void createFile(String fileName) {
30     Path f = Paths.get(fileName);
31     try {
32         if (Files.exists(f)) {
33             System.out.println("File already exists");
34         } else {
35             Path cFile = Files.createFile(f);
36             System.out.println("File created at " + cFile.toString());
37         }
38     } catch (Exception e) {
39         e.printStackTrace();
40     }
41 }
42
43 public void readFile() {
44     Path f = Paths.get(fileName);
45     try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.READ)) {
46         ByteBuffer buffer = ByteBuffer.allocate(1024);
47         while (fileChannel.read(buffer) > 0) {
48             buffer.flip();
49             while (buffer.hasRemaining()) {
50                 System.out.print((char) buffer.get());
51             }
52             buffer.clear();
53         }
54     } catch (IOException e) {
55     }
56 }
```

```
wipro - firstjava/src/com/wipro/Mnioc.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  MyTaskThread.java
  Serverpgm.java
  Stream.java
  Stream7.java
  Task.java
  User.java
  com.wipro
    BuiltInAnno.java
    CompletableFuture.java
    CustomAnno.java
    Employee.java
    ExecutorExamp.java
    FileDemo.java
    First.java
    GenericClass.java
    GenericLUB.java
    LinkedListDemo.java
    Manager.java
    Mnioc.java
    PC.java
    Salary.java
    ThreadEg2.java
    ThreadEg3.java
    TimeDateEg.java
    URLLDemo.java
  com.wipro.selectors
  pra
  practice
  wiproAssignment
    JRE System Library (JavaSE-17)
    src
      assignments.day20
        DaysBetweenDates.java
        FileCopyWithNIO.java
        JavaIO.java
        JavaNetworking.java
        Operation.java
        SerialDeserial.java

Mnioc.java
53
54 } catch (IOException e) {
55     e.printStackTrace();
56 }
57
58
59 public void writeFile(String fileName) {
60     Path f = Paths.get(fileName);
61     try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.WRITE, Stan
62         String content = "Johny Johny, Yes Papa,\nEating sugar? No Papa";
63         ByteBuffer buffer = ByteBuffer.wrap(content.getBytes());
64         fileChannel.write(buffer);
65         System.out.println("Data Written Successfully");
66     } catch (IOException e) {
67         e.printStackTrace();
68     }
69 }
70
71 public void appendFile(String fileName) {
72     Path f = Paths.get(fileName);
73     try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.APPEND)) {
74         String content = "\nTelling Lies? No Papa,\nOpen your Mouth, Ha Ha Ha :)";
75         ByteBuffer buffer = ByteBuffer.wrap(content.getBytes());
76         fileChannel.write(buffer);
77         System.out.println("Data Appended Successfully");
78     } catch (IOException e) {
79         e.printStackTrace();
80     }
81 }
82
83 public static void main(String[] args) {
84     Mnioc mn = new Mnioc();
85
86     // Create a directory
87     mn.createDirectory();
88
89     // Create a file
90     // mn.createFile("mydir/rhymes.txt");
91     System.out.println("--Writing ---");
92     // Write to a file
93     mn.writeFile(mn.fileName);
94     System.out.println("--Reading ---");
95     // Read from file
96 }
```

```
wipro - firstjava/src/com/wipro/Mnioc.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  MyTaskThread.java
  Serverpgm.java
  Stream.java
  Stream7.java
  Task.java
  User.java
  com.wipro
    BuiltInAnno.java
    CompletableFuture.java
    CustomAnno.java
    Employee.java
    ExecutorExamp.java
    FileDemo.java
    First.java
    GenericClass.java
    GenericLUB.java
    LinkedListDemo.java
    Manager.java
    Mnioc.java
    PC.java
    Salary.java
    ThreadEg2.java
    ThreadEg3.java
    TimeDateEg.java
    URLLDemo.java
  com.wipro.selectors
  pra
  practice
  wiproAssignment
    JRE System Library (JavaSE-17)
    src
      assignments.day20
        DaysBetweenDates.java
        FileCopyWithNIO.java
        JavaIO.java
        JavaNetworking.java
        Operation.java
        SerialDeserial.java

Mnioc.java
69
70
71 public void appendFile(String fileName) {
72     Path f = Paths.get(fileName);
73     try (FileChannel fileChannel = FileChannel.open(f, StandardOpenOption.APPEND)) {
74         String content = "\nTelling Lies? No Papa,\nOpen your Mouth, Ha Ha Ha :)";
75         ByteBuffer buffer = ByteBuffer.wrap(content.getBytes());
76         fileChannel.write(buffer);
77         System.out.println("Data Appended Successfully");
78     } catch (IOException e) {
79         e.printStackTrace();
80     }
81 }
82
83 public static void main(String[] args) {
84     Mnioc mn = new Mnioc();
85
86     // Create a directory
87     mn.createDirectory();
88
89     // Create a file
90     // mn.createFile("mydir/rhymes.txt");
91     System.out.println("--Writing ---");
92     // Write to a file
93     mn.writeFile(mn.fileName);
94     System.out.println("--Reading ---");
95     // Read from file
96 }
```



```
wipro - firstjava/src/com/wipro/Mnioc.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  MyTaskThread.java
  Serverpgm.java
  Stream.java
  Stream7.java
  Task.java
  User.java
  com.wipro
    BuiltInAnno.java
    CompletableFuture.java
    CustomAnno.java
    Employee.java
    ExecutorExamp.java
    FileDemo.java
    First.java
    GenericClass.java
    GenericClass.java
    GenericListDemo.java
    Manager.java
    Mnioc.java
    PC.java
    Salary.java
    ThreadEg2.java
    ThreadEg3.java
    TimeDateEg.java
    URLEDemo.java
  pra
  practice
  wiproAssignment
  JRE System Library [JavaSE-17]
  src
    assignments.day20
      DaysBetweenDates.java
      FileCopyWithNIO.java
      JavaIO.java
      JavaNetworking.java
      Operation.java
      SerialDeserial.java

Mnioc.java
78 } catch (IOException e) {
79     e.printStackTrace();
80 }
81 }
82
83 public static void main(String[] args) {
84     Mnioc mn = new Mnioc();
85
86     // Create a directory
87     mn.createDirectory();
88
89     // Create a file
90     mn.createFile("mydir/rhymes.txt");
91     System.out.println("--Writing ---");
92     // Write to a file
93     mn.writeFile(mn.fileName);
94     System.out.println("--Reading ---");
95     // Read from file
96     mn.readFile();
97     System.out.println("--Appending ---");
98     // Append to a file
99     mn.appendFile(mn.fileName);
100    System.out.println("--Read after append ---");
101    // Read from file
102    mn.readFile();
103 }
104 }
```

```
wipro - firstjava/src/com/wipro/Mnioc.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

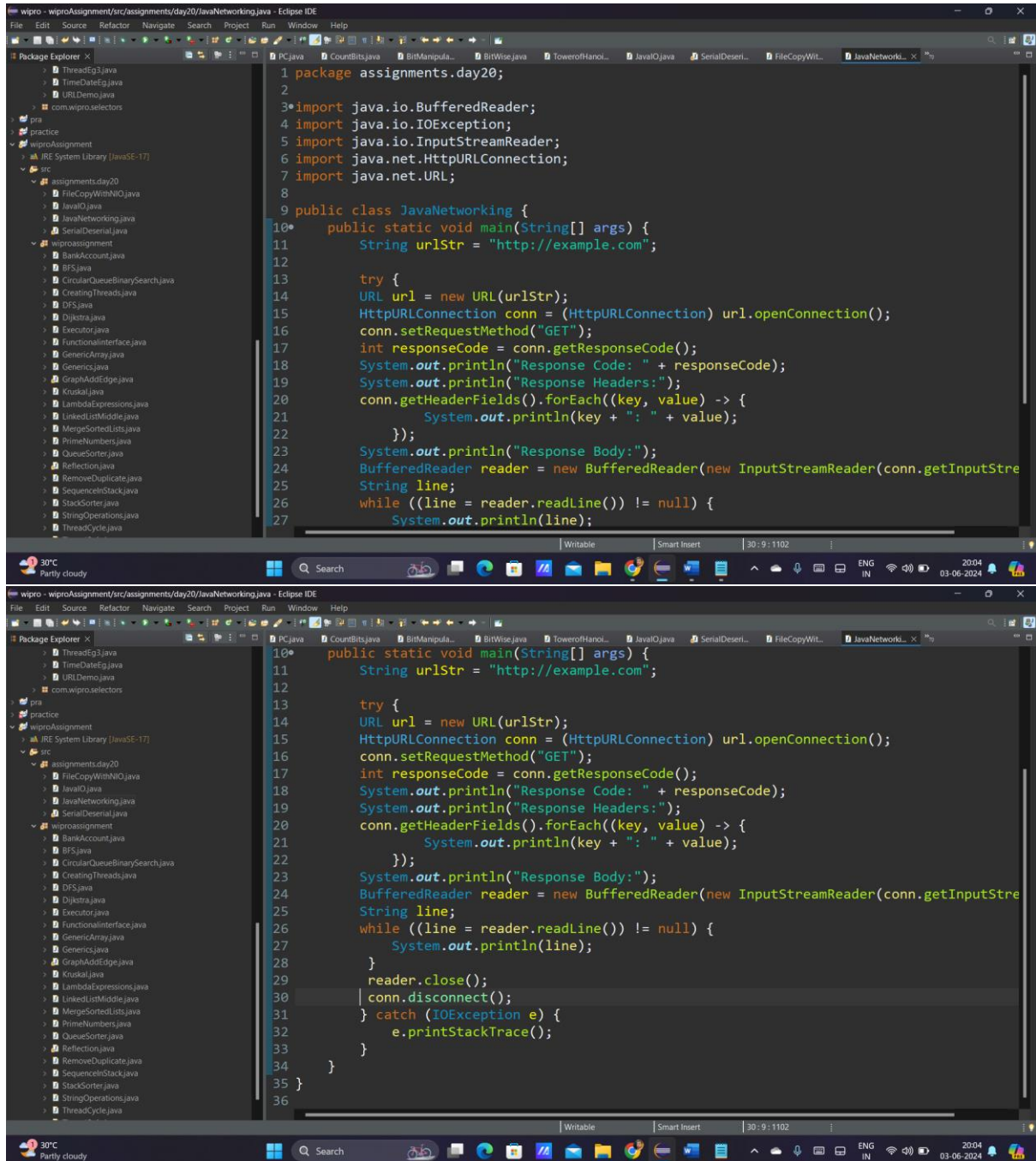
Package Explorer
  MyTaskThread.java
  Serverpgm.java
  Stream.java
  Stream7.java
  Task.java
  User.java
  com.wipro
    BuiltInAnno.java
    CompletableFuture.java
    CustomAnno.java
    Employee.java
    ExecutorExamp.java
    FileDemo.java
    First.java
    GenericClass.java
    GenericClass.java
    GenericListDemo.java
    Manager.java
    Mnioc.java
    PC.java
    Salary.java
    ThreadEg2.java
    ThreadEg3.java
    TimeDateEg.java
    URLEDemo.java
  pra
  practice
  wiproAssignment
  JRE System Library [JavaSE-17]
  src
    assignments.day20
      DaysBetweenDates.java
      FileCopyWithNIO.java
      JavaIO.java
      JavaNetworking.java
      Operation.java
      SerialDeserial.java

Mnioc.java
1 package com.wipro;
2
3 import java.io.IOException;
4
5
10
11
12 public class Mnioc {
13     String fileName = "mydir/rhymes.txt";
14
15     public void createDirectory() {
16         Path p = Paths.get("mydir");
17         try {
18             if (Files.exists(p)) {
19                 System.out.println("Direc
20             } else {
21                 Path cPath = Files.create
22                 System.out.println("Direc
23             }
24         } catch (Exception e) {
25             e.printStackTrace();
26         }
27     }
28
29     public void createFile(String fileName) {
30         Path f = Paths.get(fileName);
31         try {
32             if (Files.exists(f)) {
33                 System.out.println("File

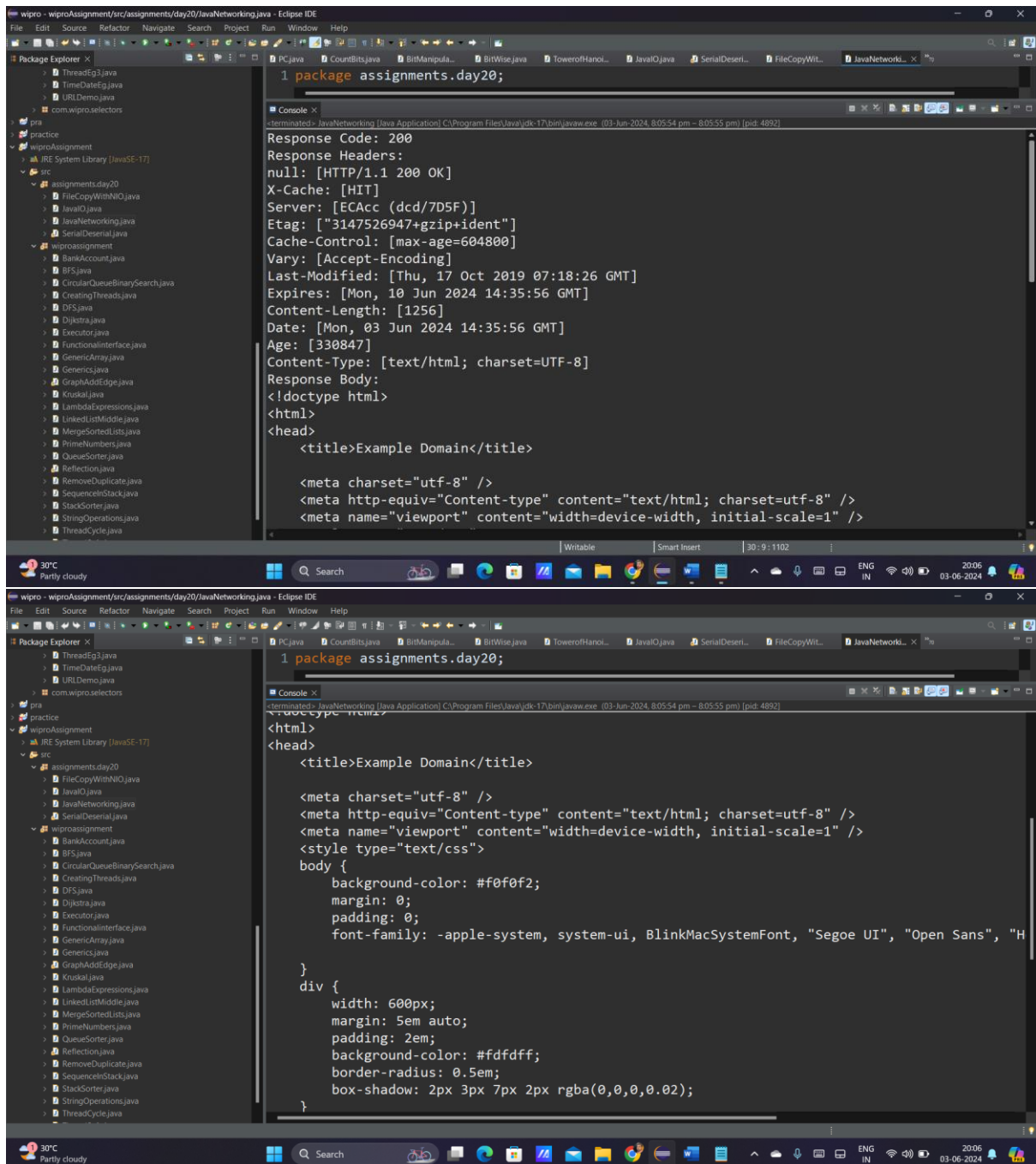
Console
<terminated> Mnioc [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (0) - Jun-2024, 9:09:01 p
Directory already exists
--Writing ---
Data Written Successfully
--Reading ---
Johnny Johnny, Yes Papa,
Eating sugar? No Papa
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)--Appending ---
Data Appended Successfully
--Read after append ---
Johnny Johnny, 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 :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
```

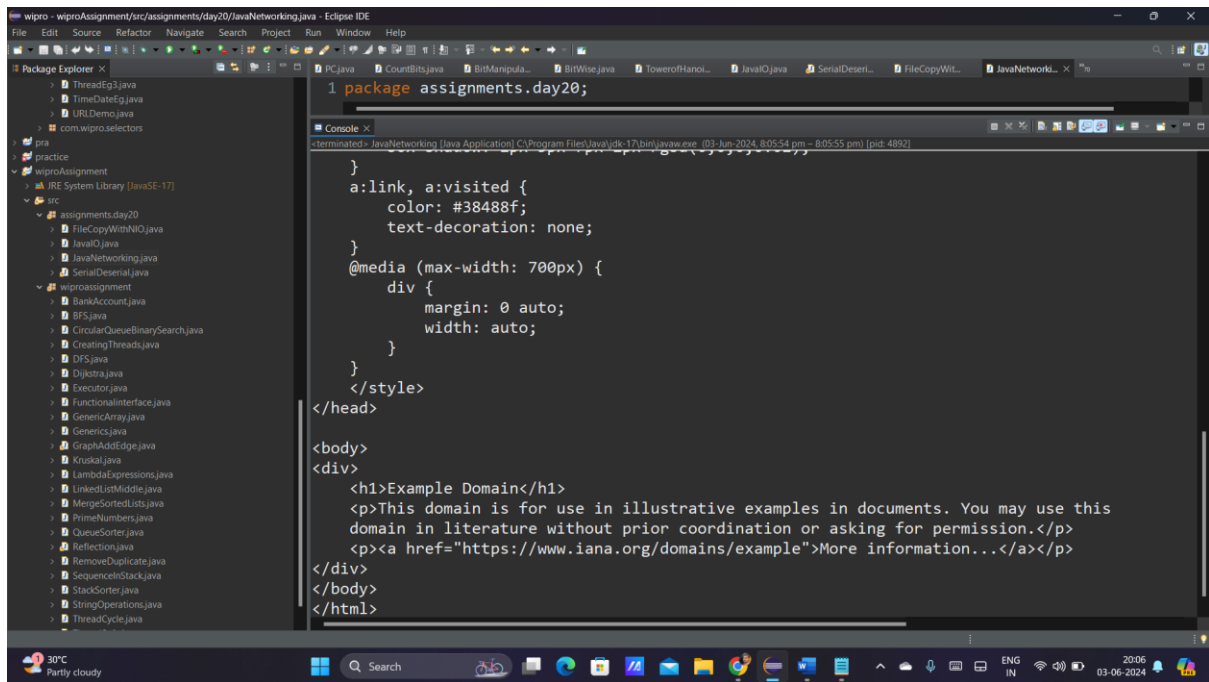
Task 4: Java Networking

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



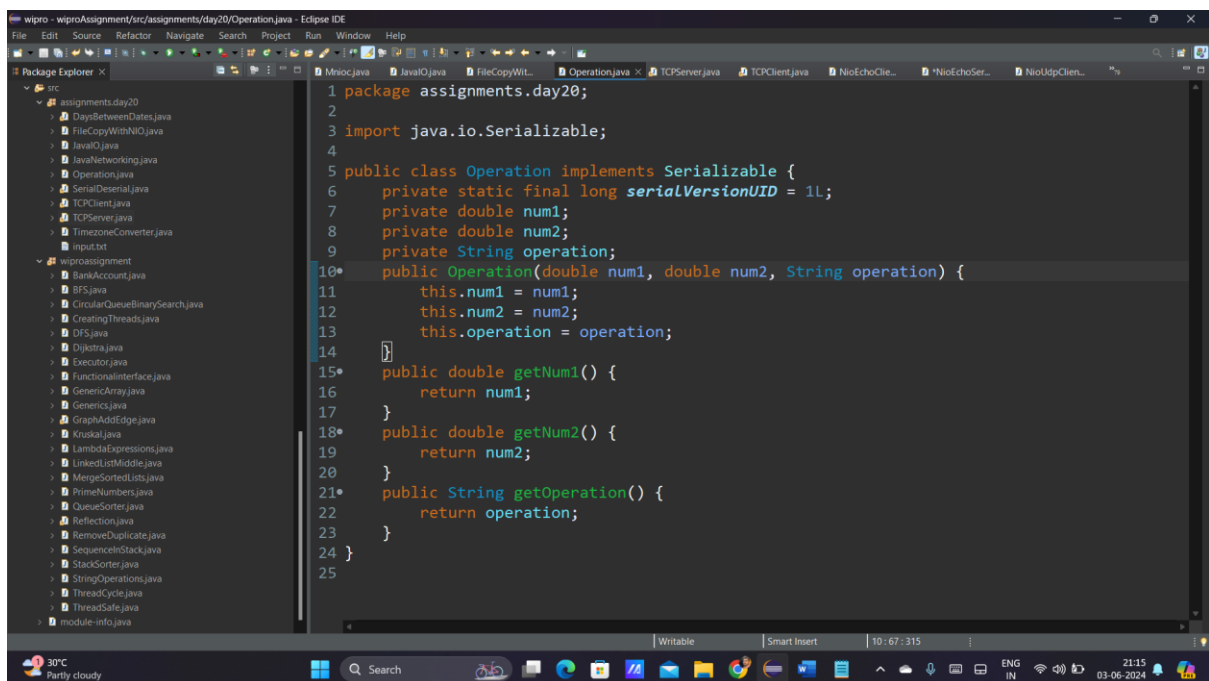
```
1 package assignments.day20;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.net.HttpURLConnection;
7 import java.net.URL;
8
9 public class JavaNetworking {
10     public static void main(String[] args) {
11         String urlStr = "http://example.com";
12
13         try {
14             URL url = new URL(urlStr);
15             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
16             conn.setRequestMethod("GET");
17             int responseCode = conn.getResponseCode();
18             System.out.println("Response Code: " + responseCode);
19             System.out.println("Response Headers:");
20             conn.getHeaderFields().forEach((key, value) -> {
21                 System.out.println(key + ": " + value);
22             });
23             System.out.println("Response Body:");
24             BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
25             String line;
26             while ((line = reader.readLine()) != null) {
27                 System.out.println(line);
28             }
29             reader.close();
30             conn.disconnect();
31         } catch (IOException e) {
32             e.printStackTrace();
33         }
34     }
35 }
36
```

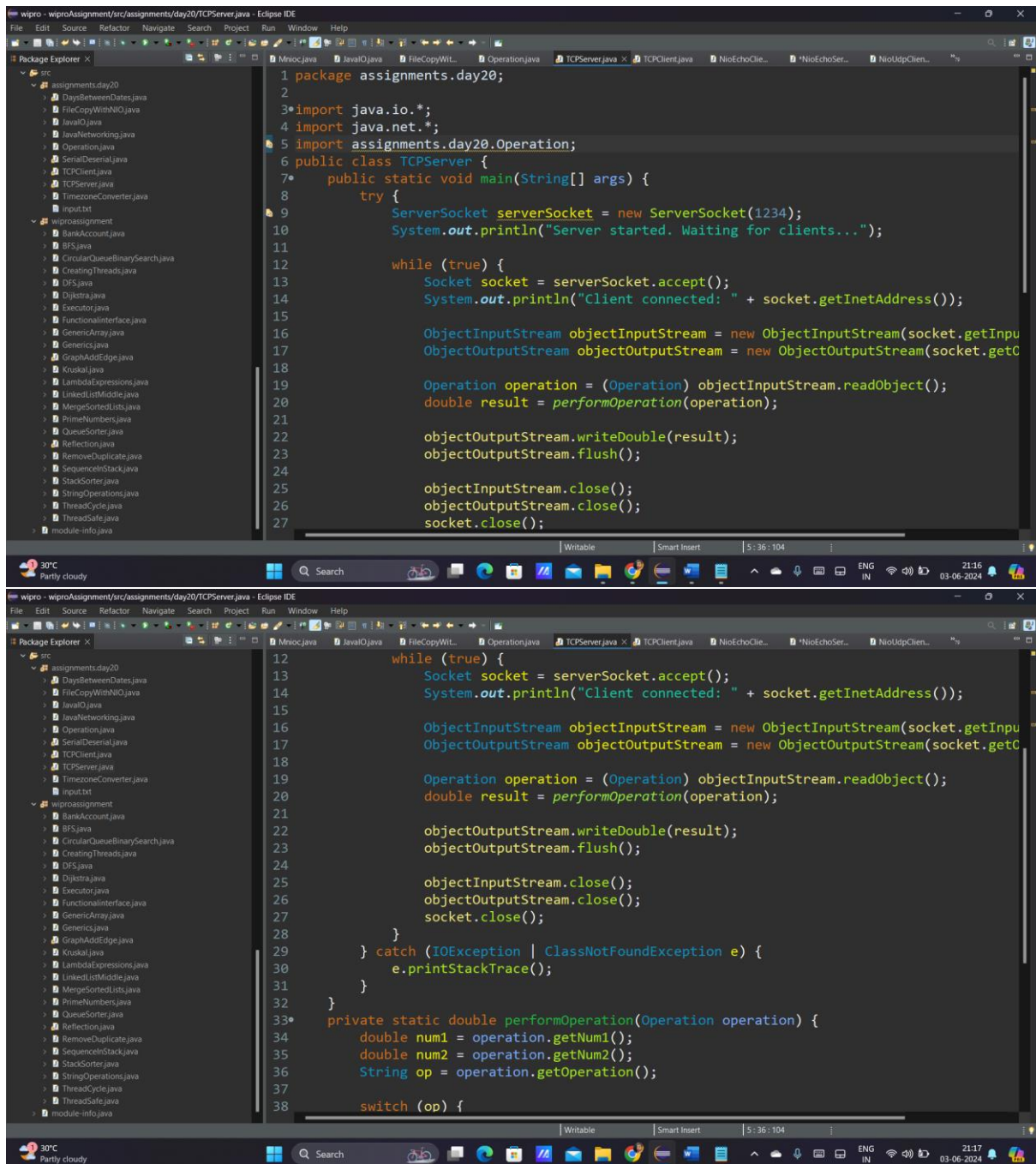



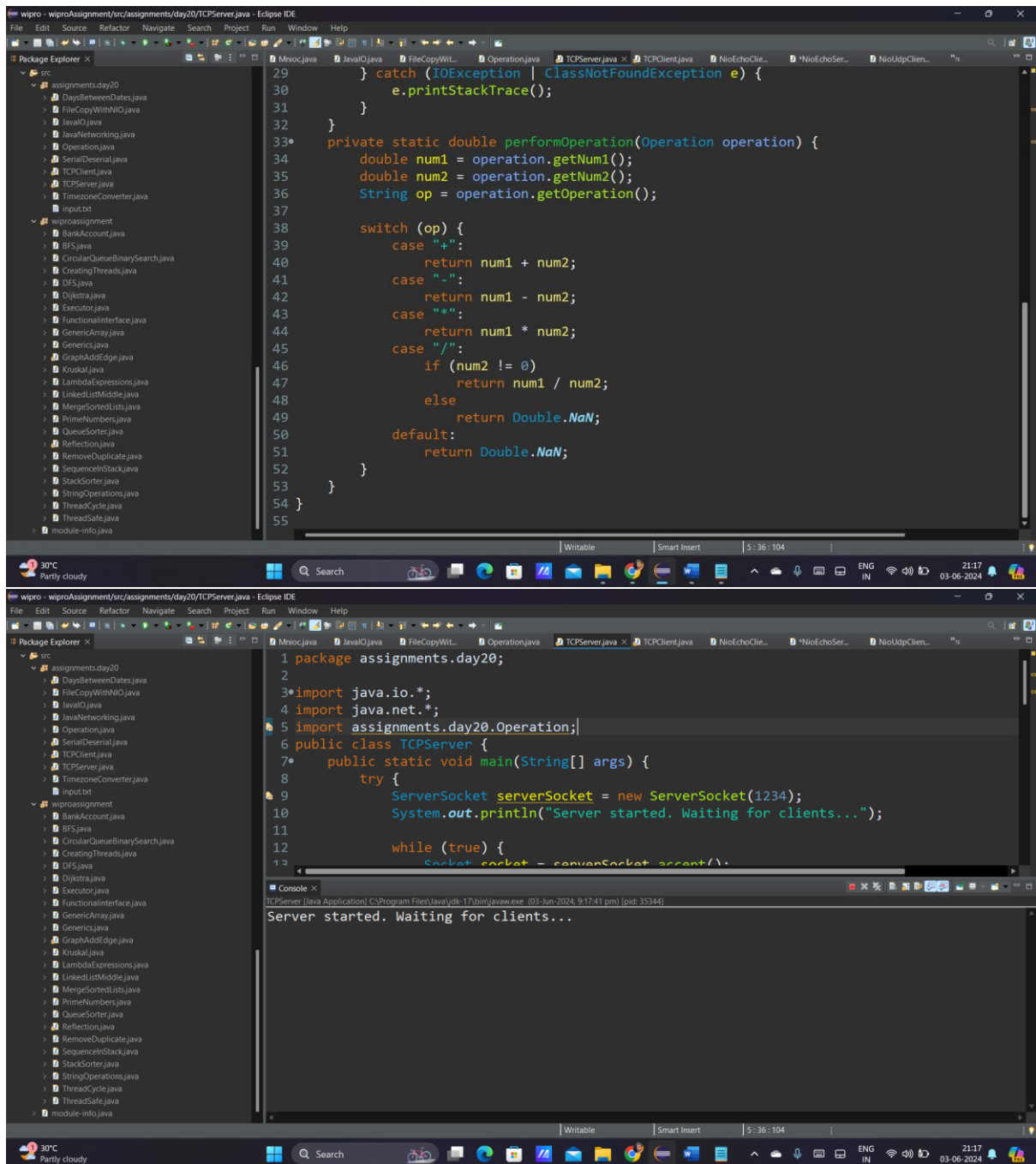


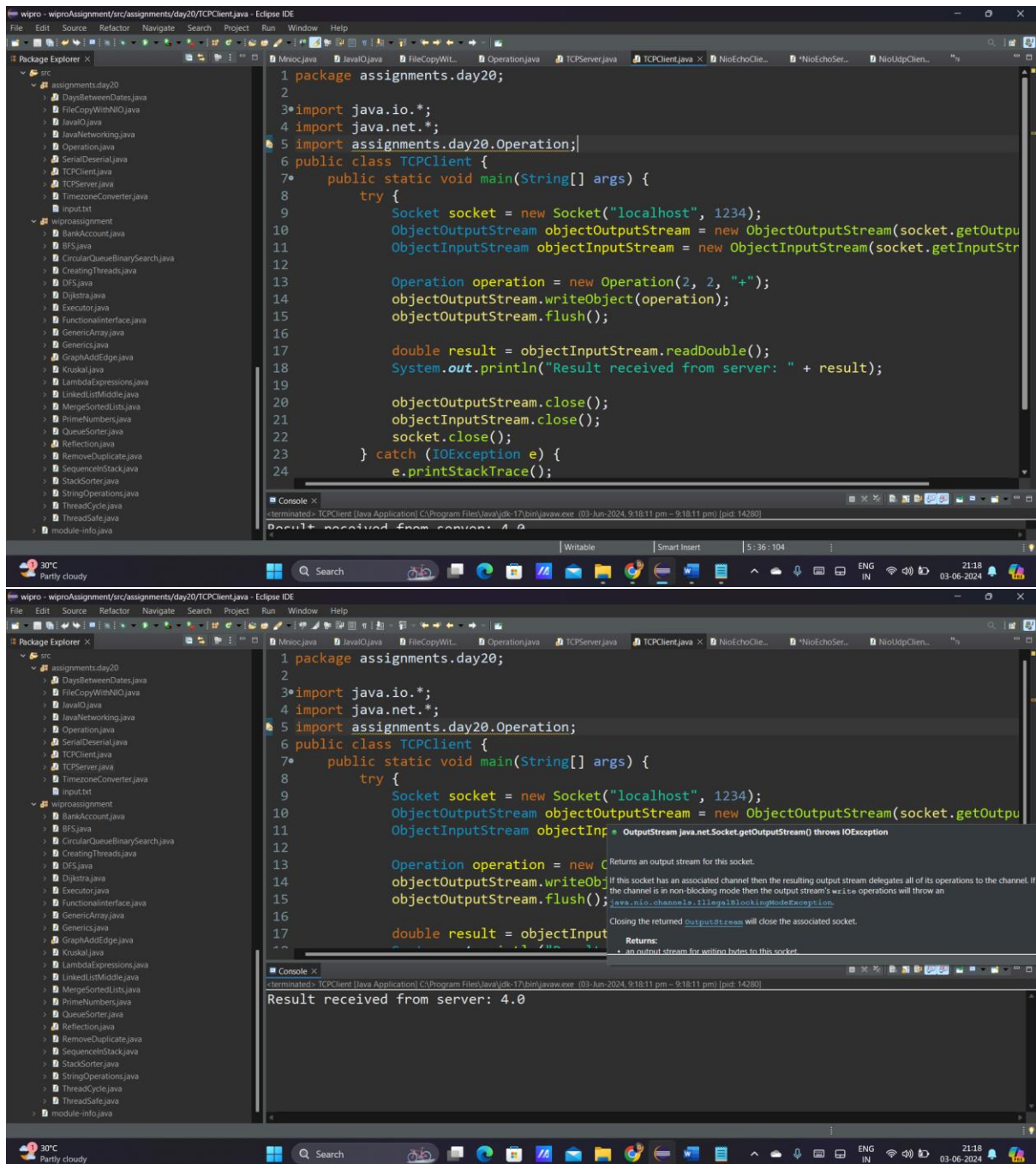
Task 5: Java Networking and Serialization

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



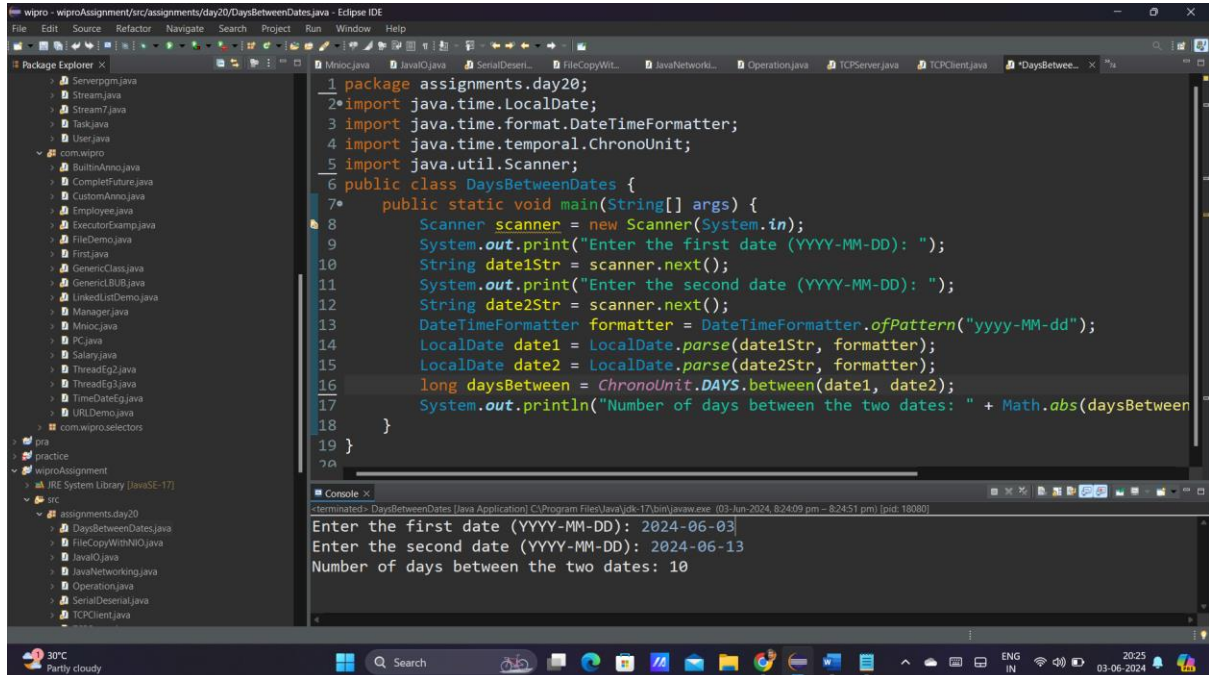






Task 6: Java 8 Date and Time API

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



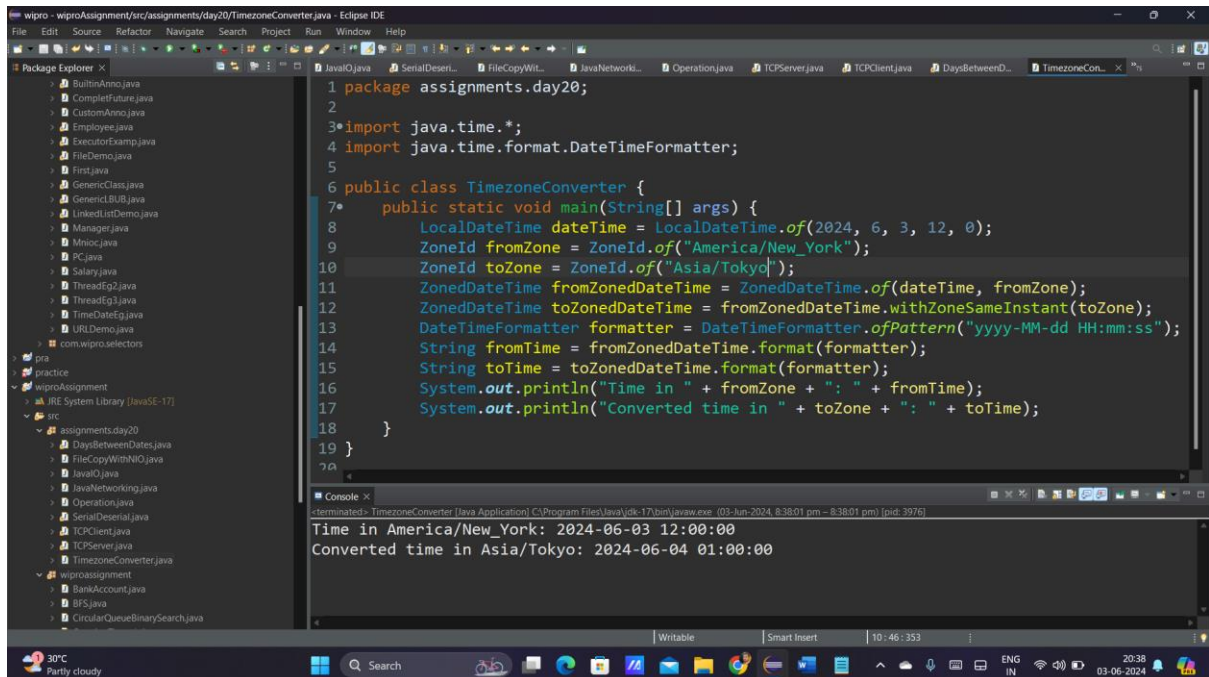
```
1 package assignments.day20;
2 import java.time.LocalDate;
3 import java.time.format.DateTimeFormatter;
4 import java.time.temporal.ChronoUnit;
5 import java.util.Scanner;
6 public class DaysBetweenDates {
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.print("Enter the first date (YYYY-MM-DD): ");
10        String date1Str = scanner.next();
11        System.out.print("Enter the second date (YYYY-MM-DD): ");
12        String date2Str = scanner.next();
13        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
14        LocalDate date1 = LocalDate.parse(date1Str, formatter);
15        LocalDate date2 = LocalDate.parse(date2Str, formatter);
16        long daysBetween = ChronoUnit.DAYS.between(date1, date2);
17        System.out.println("Number of days between the two dates: " + Math.abs(daysBetween));
18    }
19 }
```

Console Output:

```
Enter the first date (YYYY-MM-DD): 2024-06-03
Enter the second date (YYYY-MM-DD): 2024-06-13
Number of days between the two dates: 10
```

Task 7: Timezone

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



```
1 package assignments.day20;
2
3 import java.time.*;
4 import java.time.format.DateTimeFormatter;
5
6 public class TimezoneConverter {
7     public static void main(String[] args) {
8         LocalDateTime dateTime = LocalDateTime.of(2024, 6, 3, 12, 0);
9         ZoneId fromZone = ZoneId.of("America/New_York");
10        ZoneId toZone = ZoneId.of("Asia/Tokyo");
11        ZonedDateTime fromZonedDateTime = ZonedDateTime.of(dateTime, fromZone);
12        ZonedDateTime toZonedDateTime = fromZonedDateTime.withZoneSameInstant(toZone);
13        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
14        String fromTime = fromZonedDateTime.format(formatter);
15        String toTime = toZonedDateTime.format(formatter);
16        System.out.println("Time in " + fromZone + ": " + fromTime);
17        System.out.println("Converted time in " + toZone + ": " + toTime);
18    }
19 }
```

Console Output:

```
Time in America/New_York: 2024-06-03 12:00:00
Converted time in Asia/Tokyo: 2024-06-04 01:00:00
```