Day-20

Name: Amit Kumar Parhi

Email:amitkparhi07@gmail.com

Task 1: Java IO Basics

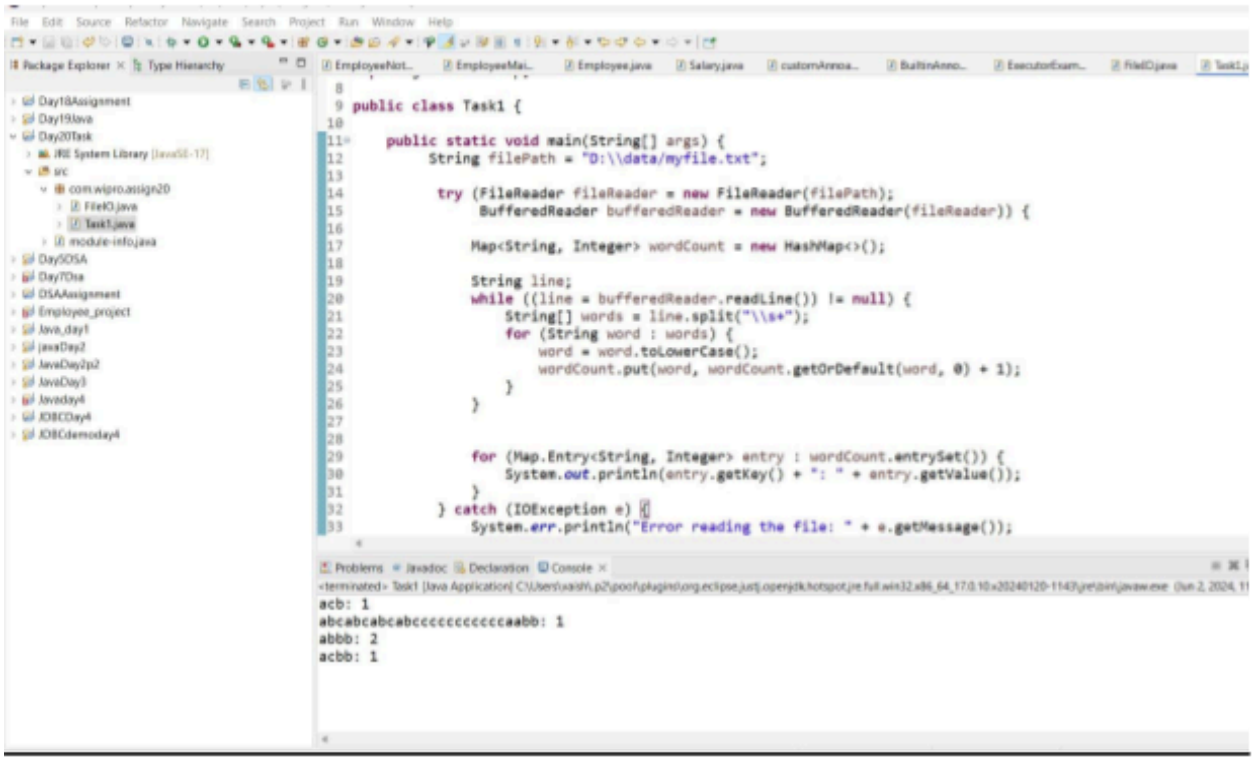
Write a program that reads a text file and counts the frequencyof each word using FileReader and FileWriter.

Solution:

```
 package com.wipro.assign20;

 import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap; import
java.util.Map;

 public class Task1 {
     public static void main(String[] args) {
      String filePath = "D:\\data/myfile.txt";

            try (FileReader fileReader = new FileReader(filePath);

BufferedReader bufferedReader = new BufferedReader(fileReader)) {

               Map<String, Integer> wordCount = new HashMap<>();
                String line;
                while ((line = bufferedReader.readLine()) != null) {
                   String[] words = line.split("\\s+");
                   for (String word : words) {
word = word.toLowerCase();
wordCount.put(word,
 wordCount.getOrDefault(word, 0) + 1);
                    }
                }



for (Map.Entry<String, Integer> entry : wordCount.entrySet()) {

   System.out.println(entry.getKey() + ": " + entry.getValue());
                }
            } catch (IOException e) {
```

```
                System.err.println("Error reading the file: " + e.getMessage());
            }

        }

}
```

Output:



Task 2: Serialization and Deserialization

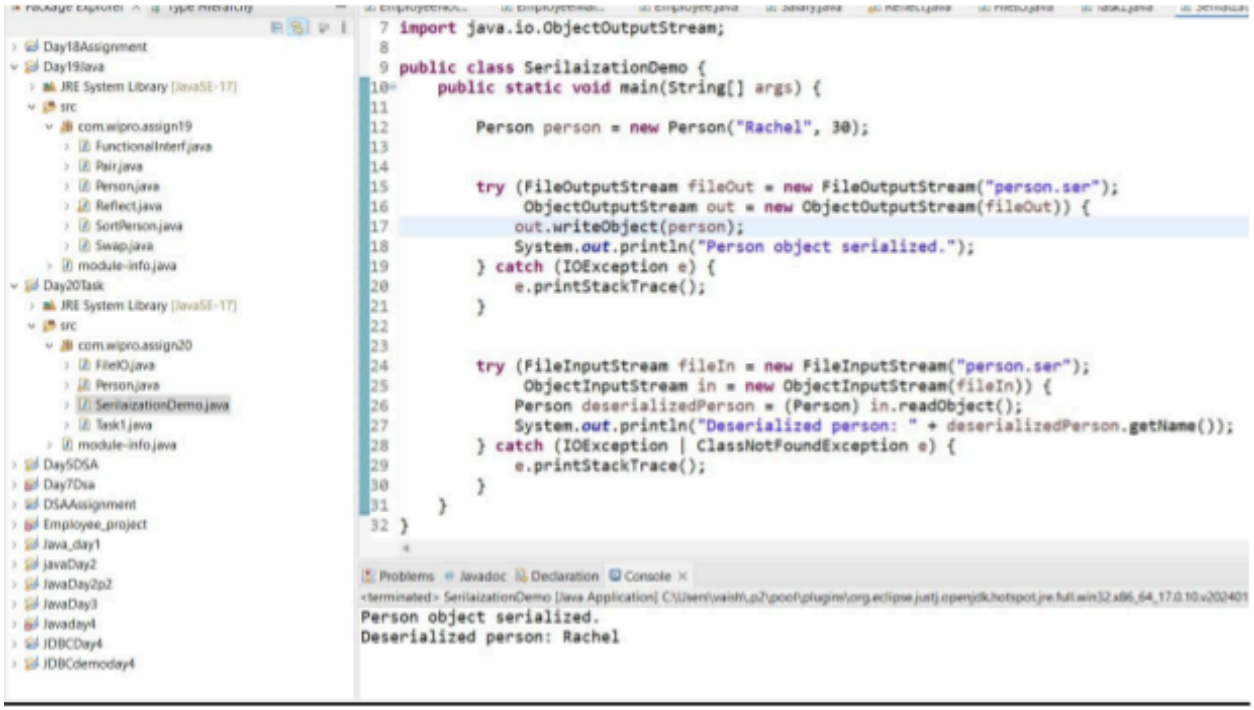
Serialize a custom object to a file and then deserialize it back to recover the object state.

Solution:

```
package com.wipro.assign20;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException; import
java.io.ObjectInputStream; import
java.io.ObjectOutputStream;

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

        Person person = new Person("Rachel", 30);
```

```
        try (FileOutputStream fileOut = new
 FileOutputStream("person.ser");
             ObjectOutputStream out = new ObjectOutputStream(fileOut))
 {
            out.writeObject(person);
            System.out.println("Person object serialized.");
} catch (IOException e) {
            e.printStackTrace();
        }

        try (FileInputStream fileIn = new
 FileInputStream("person.ser");
             ObjectInputStream in = new ObjectInputStream(fileIn)) {
            Person deserializedPerson = (Person) in.readObject();
System.out.println("Deserialized person: " + deserializedPerson.getName());
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

Output:



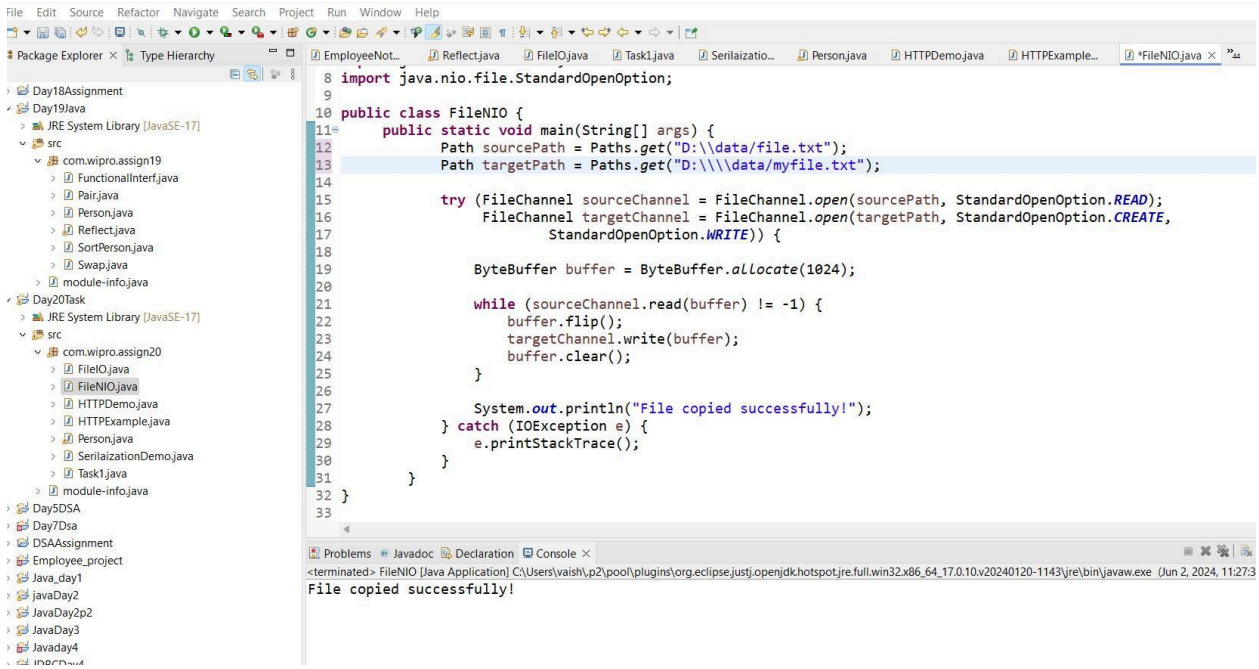
Task 3: New IO (NIO) Use NIO Channels and Buffers to read content from a file and write to another file.

Solution:

```
package com.wipro.assign20;
```

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

 public class FileNIO {
       public static void main(String[] args) {
              Path sourcePath = Paths.get("D:\\data/file.txt");
              Path targetPath = Paths.get("D:\\\\data/myfile.txt");
             try (FileChannel sourceChannel =
FileChannel.open(sourcePath, StandardOpenOption.READ);
                    FileChannel targetChannel =
 FileChannel.open(targetPath, StandardOpenOption.CREATE,
                           StandardOpenOption.WRITE)) {

                  ByteBuffer buffer = ByteBuffer.allocate(1024);

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

                 System.out.println("File copied successfully!");
             } catch (IOException e) {
e.printStackTrace();
              }
          }
 }
```

Output:



Task 4: Java Networking

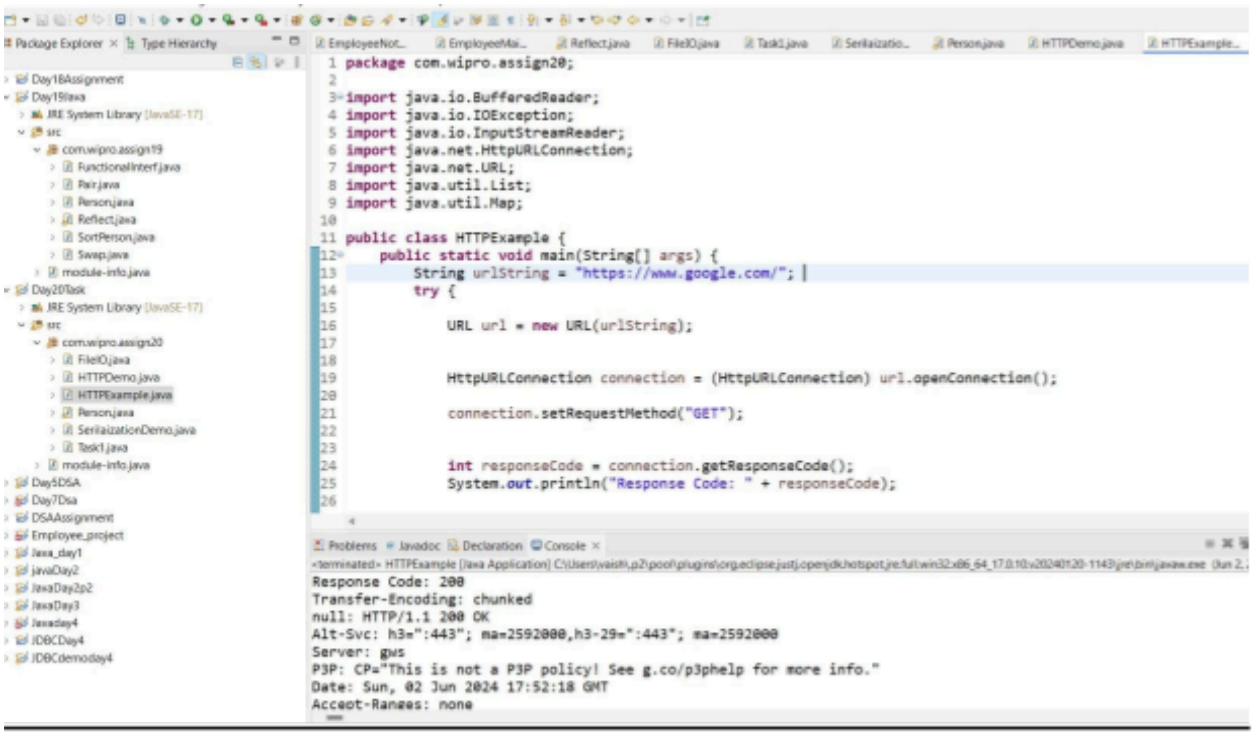
Write a simple HTTP client that connects to a URL, sends a request, and displays the response headers and body.

Solution:

```
package com.wipro.assign20;

import java.io.BufferedReader;
import java.io.IOException; import
java.io.InputStreamReader; import
java.net.HttpURLConnection; import
java.net.URL; import
java.util.List; import
java.util.Map;

public class HTTPExample {     public static void
main(String[] args) {          String urlString =
"https://www.google.com/";          try {

            URL url = new URL(urlString);


            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();

            connection.setRequestMethod("GET");
int responseCode = connection.getResponseCode();
```

```
System.out.println("Response Code: " +
responseCode);


            Map<String, List<String>> headers =
connection.getHeaderFields();             for
(Map.Entry<String, List<String>> entry :
headers.entrySet()) {
                String headerName = entry.getKey();
for (String value : entry.getValue()) {
                    System.out.println(headerName + ": " + value);
                }
            }


            BufferedReader in = new BufferedReader(new
 InputStreamReader(connection.getInputStream()));
            String inputLine;
            StringBuilder responseBody = new StringBuilder();
while ((inputLine = in.readLine()) != null) {
responseBody.append(inputLine);
            }
in.close();

            System.out.println("Response Body:");
            System.out.println(responseBody.toString());

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

}
```

Output:



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

Solution:

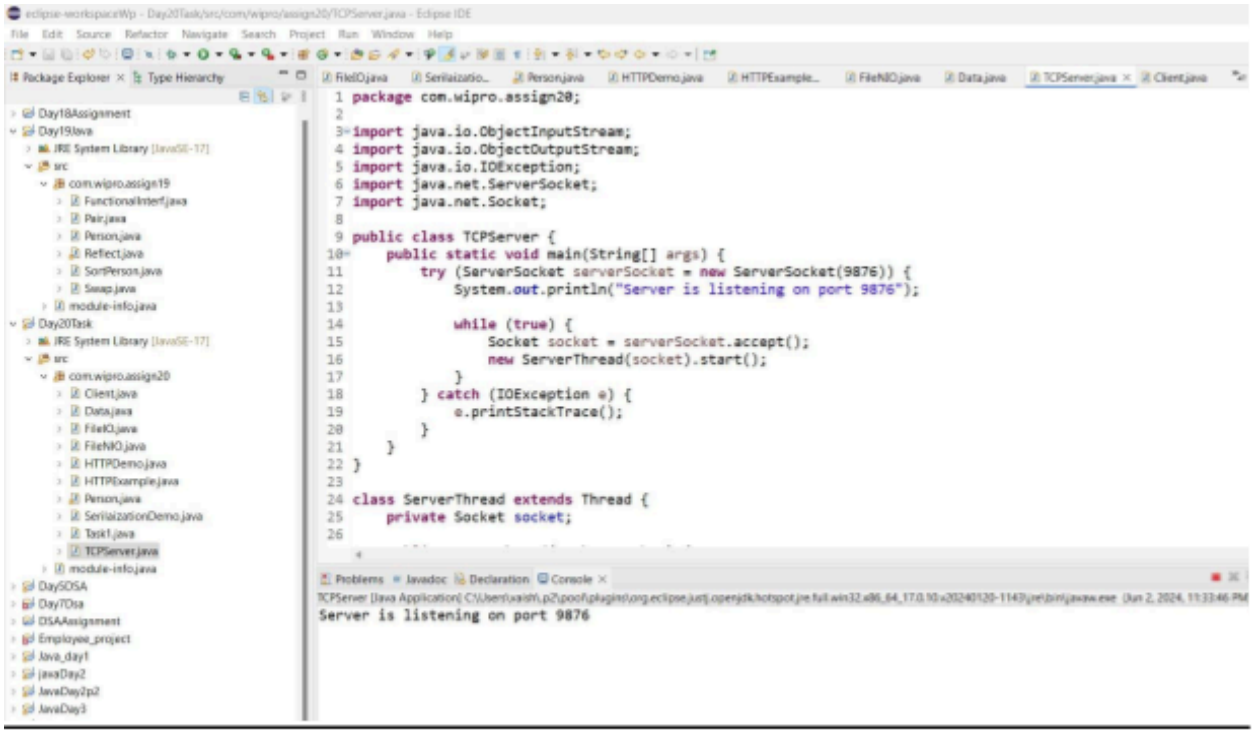
```
package com.wipro.assign20;

import java.io.Serializable;

public class Data implements Serializable {
private static final long serialVersionUID = 1L;
private int number1;     private int number2;
private String operation;

    public Dat (int number1, int number2, String operation) {
           a
        this.number1 = number1;
        this.number2 = number2;
this.operation = operation;
     }
```

```
    public int getNumber1() {
return number1;
    }
    public int getNumber2() {
return number2;
    }
    public String getOperation() {
return operation;
    }
 }


 package com.wipro.assign20;

 import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.IOException; import
java.net.ServerSocket; import
java.net.Socket;

 public class TCPServer {
    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(9876)) {
System.out.println("Server is listening on port 9876");

            while (true) {
                Socket socket = serverSocket.accept();
new ServerThread(socket).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
 }
 class ServerThread extends Thread {
private Socket socket;

    public ServerThread(Socket socket) {
this.socket = socket;
    }
    public void run() {
        try (ObjectInputStream input = new
ObjectInputStream(socket.getInputStream());
             ObjectOutputStream output = new
 ObjectOutputStream(socket.getOutputStream())) {
            Data data = (Data) input.readObject();
int result = performOperation(data);
```

```
            output.writeObject(result);
output.flush();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
    private int performOperation(Data data) {
int number1 = data.getNumber1();         int
number2 = data.getNumber2();         String
operation = data.getOperation();

        switch (operation) {
case "+":
                return number1 + number2;
case "-":
                return number1 - number2;
case "*":
                return number1 * number2;
case "/":                   if (number2 != 0) {
return number1 / number2;
                } else {
                    throw new ArithmeticException("Division by zero");
                }             default:                  throw new
UnsupportedOperationException("Unsupported operation: " +
operation);
        }
    }
 }



 package com.wipro.assign20;

 import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;

 public class Client {
    public static void main(String[] args)
{         String hostname = "localhost";
int port = 9876;

        try (Socket socket = new Socket(hostname, port);
             ObjectOutputStream output = new
```

```
 ObjectOutputStream(socket.getOutputStream());
              ObjectInputStream input = new
 ObjectInputStream(socket.getInputStream())) {

             // Send operation data to the server
Data data = new Data(2, 2, "+");
output.writeObject(data);              output.flush();

            // Receive the result from the server
int result = (int) input.readObject();
System.out.println("Result: " + result);

         } catch (IOException | ClassNotFoundException e) {
             e.printStackTrace();
         }
     }
 }
```

Output:



Task 6: Java 8 Date and Time API

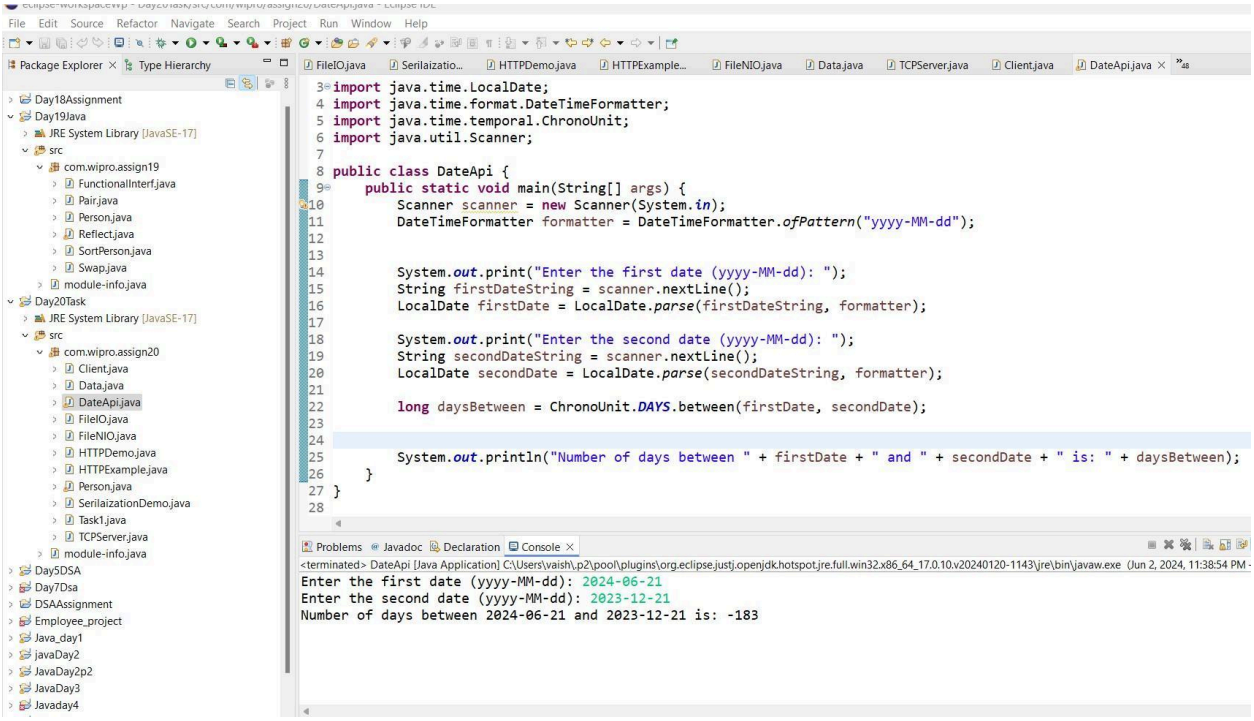
Write a program that calculates the number of days between two dates input by the user.

Solution:

```
package com.wipro.assign20;
```

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

 public class DateApi {
     public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
         DateTimeFormatter formatter =
 DateTimeFormatter.ofPattern("yyyy-MM-dd");


         System.out.print("Enter the first date (yyyy-MM-dd): ");
         String firstDateString = scanner.nextLine();
LocalDate firstDate = LocalDate.parse(firstDateString,
formatter);

         System.out.print("Enter the second date (yyyy-MM-dd): ");
String secondDateString = scanner.nextLine();
         LocalDate secondDate =
LocalDate.parse(secondDateString, formatter);

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


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

Output:



Task 7: Timezone

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

Solution:

```
package com.wipro;
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
public class TimezoneConverter {
   public static String convertTimeZone(String dateTime, String sourceTimeZone, String
targetTimeZone) {

       DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
              LocalDateTime localDateTime = LocalDateTime.parse(dateTime, formatter);

       ZoneId sourceZoneId = ZoneId.of(sourceTimeZone);
       ZoneId targetZoneId = ZoneId.of(targetTimeZone);

       ZonedDateTime sourceZonedDateTime = ZonedDateTime.of(localDateTime, sourceZoneId);

       ZonedDateTime targetZonedDateTime = sourceZonedDateTime.withZoneSameInstant(targetZoneId);

       return targetZonedDateTime.format(formatter);
   }
   public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);
       System.out.print("Enter date and time (yyyy-MM-dd HH:mm:ss): ");
       String dateTime = scanner.nextLine();
```

```
        System.out.print("Enter source time zone (e.g., America/New_York): ");
        String sourceTimeZone = scanner.nextLine();
        System.out.print("Enter target time zone (e.g., Europe/London): ");
        String targetTimeZone = scanner.nextLine();
        String result = convertTimeZone(dateTime, sourceTimeZone, targetTimeZone);
        System.out.println("Converted date and time: " + result);
        scanner.close();
    }
}
```

Output:

```
1  package com.wipro;
2
3  import java.time.LocalDateTime;
4  import java.time.ZoneId;
5  import java.time.ZonedDateTime;
6  import java.time.format.DateTimeFormatter;
7  import java.util.Scanner;
8
9  public class TimezoneConverter {
10
11     public static String convertTimeZone(String dateTime, String sourceTimeZone, String targetTimeZone) {
12
13         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
14
15
16         LocalDateTime localDateTime = LocalDateTime.parse(dateTime, formatter);
17
18
19         ZoneId sourceZoneId = ZoneId.of(sourceTimeZone);
20         ZoneId targetZoneId = ZoneId.of(targetTimeZone);
21
22
23         ZonedDateTime sourceZonedDateTime = ZonedDateTime.of(localDateTime, sourceZoneId);
24
25
26         ZonedDateTime targetZonedDateTime = sourceZonedDateTime.withZoneSameInstant(targetZoneId);
27
28
29         return targetZonedDateTime.format(formatter);
30     }
31
```

Problems Javadoc Declaration Console

```
<terminated> okkk [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (Jun 4, 2024, 10:06:25 AM – 10:07:10 AM) [pid: 4204]
Enter date and time (yyyy-MM-dd HH:mm:ss): 2024-06-04 15:00:00
Enter source time zone (e.g., America/New_York): America/New_York
Enter target time zone (e.g., Europe/London): Europe/London
Converted date and time: 2024-06-04 20:00:00
```