Name:vaishnavi Ingole

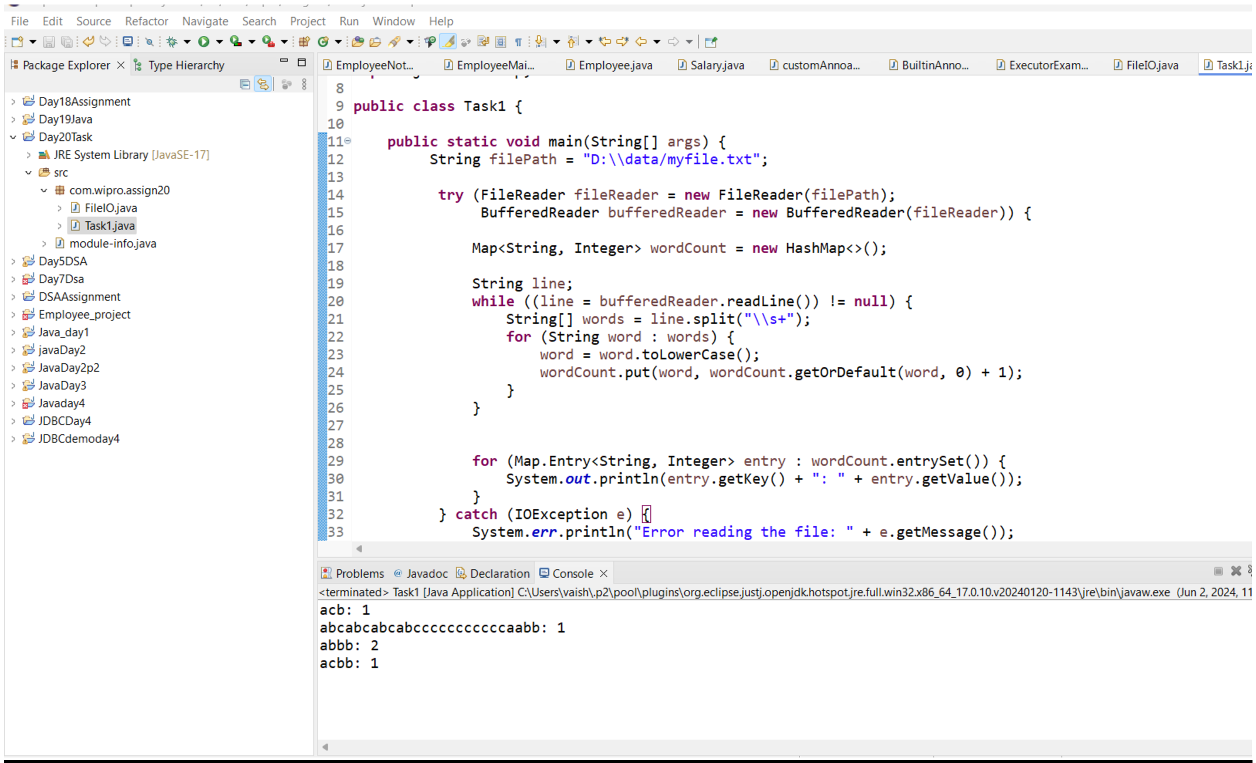
Mail:vaishnavingole54@gmail.com

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.

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());
```

```
            }
        }
}
```



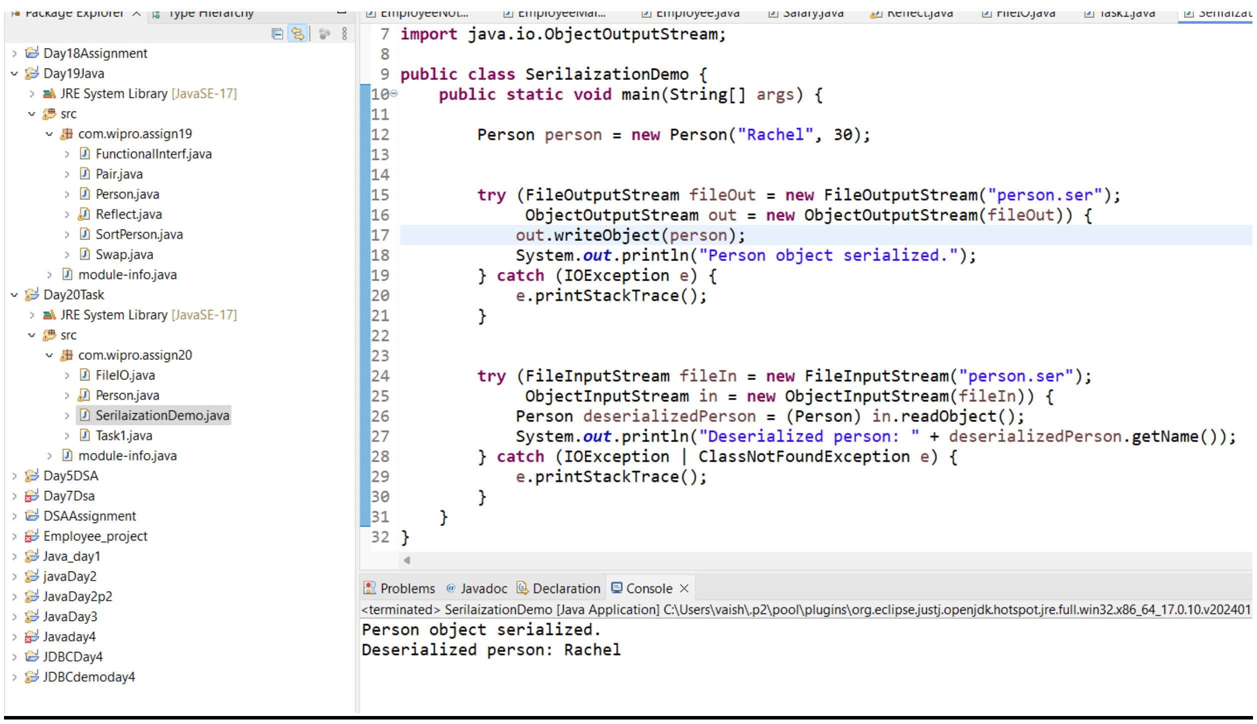
```
public class Task1 {

    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());
```

```
<terminated> Task1 [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (Jun 2, 2024, 11
acb: 1
abcabcabcabcccccccccccaabb: 1
abbb: 2
acbb: 1
```

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();
        }
    }
}
```



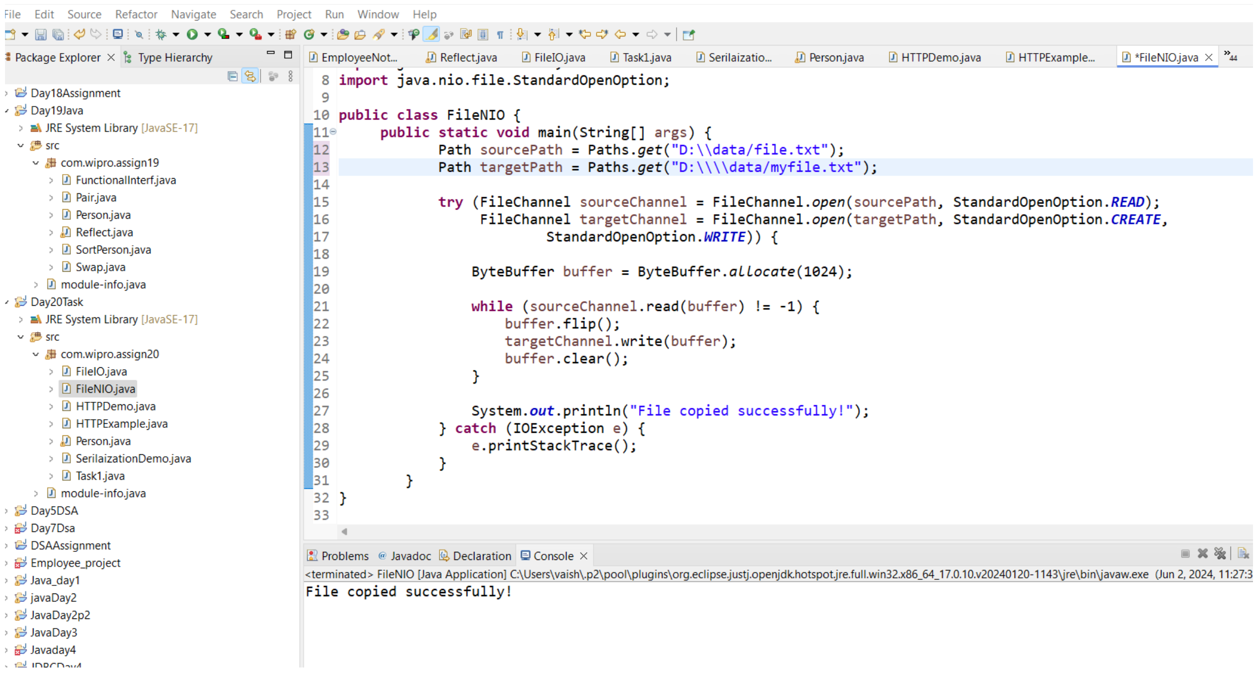
```
import java.io.ObjectOutputStream;

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();
        }
    }
}
```

```
<terminated> SerilaizationDemo [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v202401
Person object serialized.
Deserialized person: Rachel
```

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();
		}
	    }
}
```


```java
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();
		}
	}
}
```

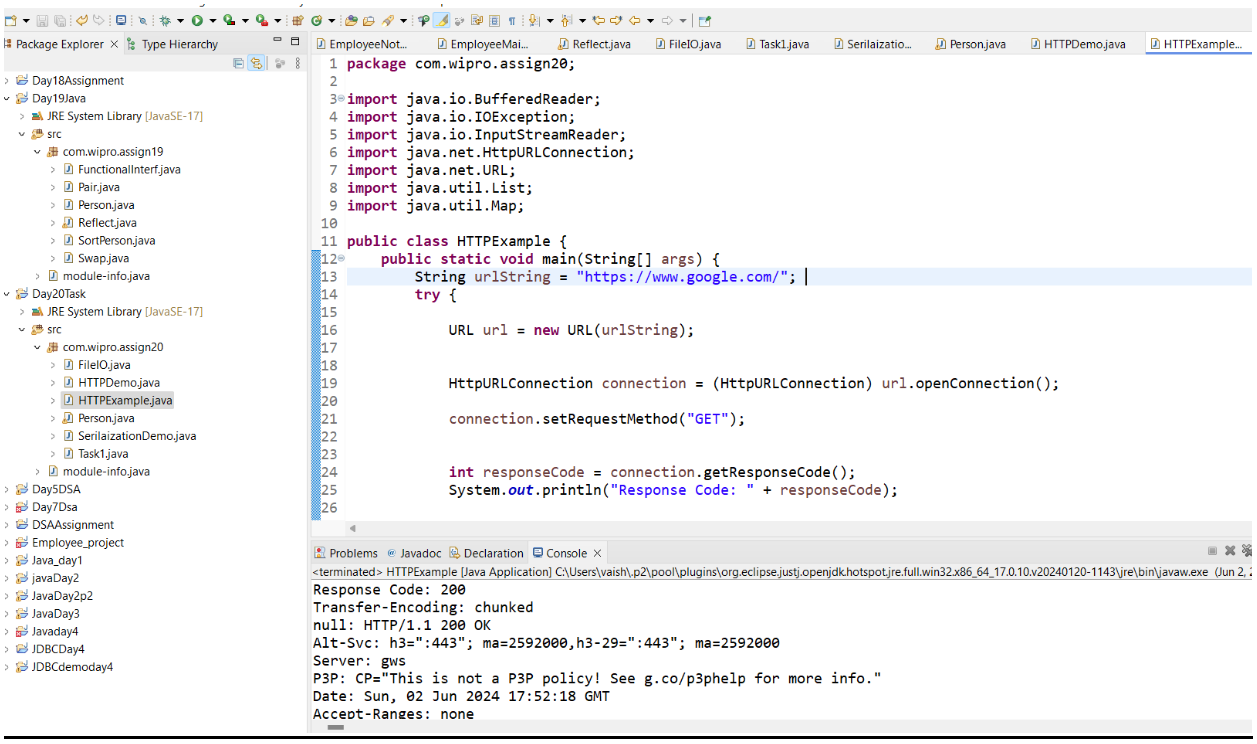
```
<terminated> FileNIO [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (Jun 2, 2024, 11:27:3
File copied successfully!
```

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:

```java
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();
        }
    }

}
```



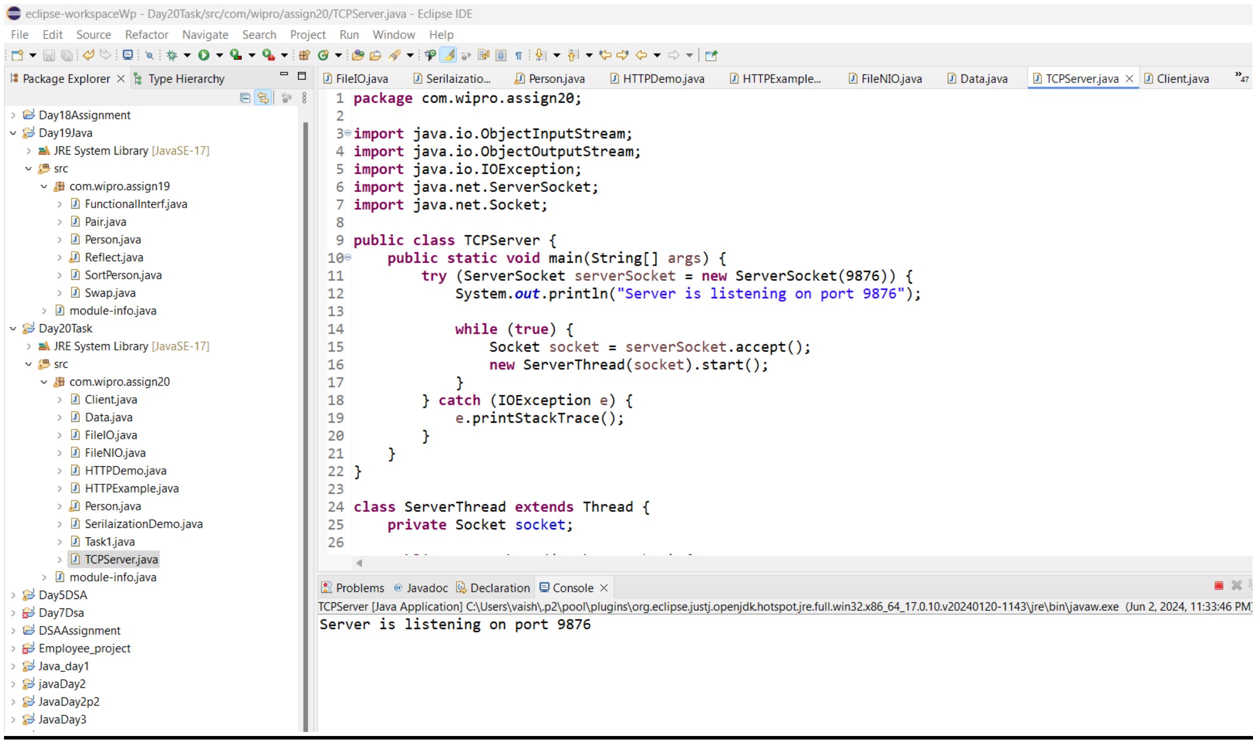
```java
package com.wipro.assign20;

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);

```

```
<terminated> HTTPExample [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Jun 2,
Response Code: 200
Transfer-Encoding: chunked
null: HTTP/1.1 200 OK
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Server: gws
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Date: Sun, 02 Jun 2024 17:52:18 GMT
Accept-Ranges: none
```

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:

```java
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 Data(int number1, int number2, String operation) {
        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();
        }
    }
}
```



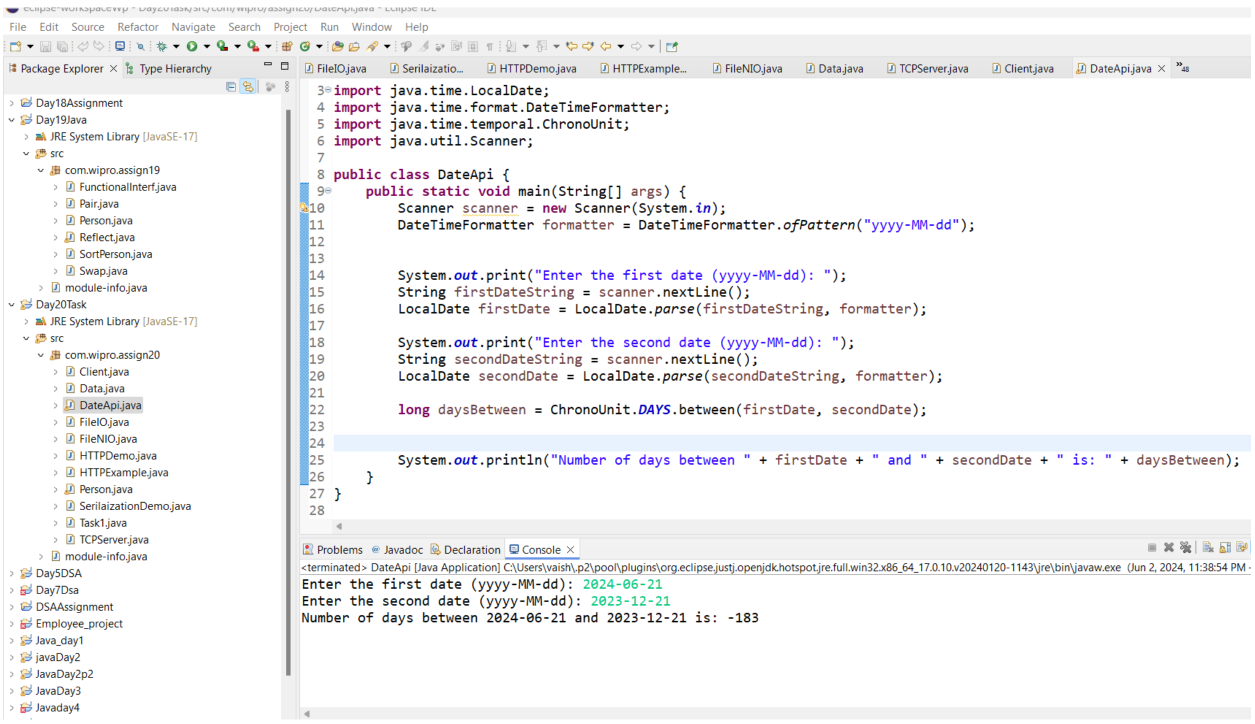
```
package com.wipro.assign20;

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;

```

```
TCPServer [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (Jun 2, 2024, 11:33:46 PM)
Server is listening on port 9876
```

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);
    }
}
```



```
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);
    }
}
```

```
<terminated> DateApi [Java Application] C:\Users\vaish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (Jun 2, 2024, 11:38:54 PM -
Enter the first date (yyyy-MM-dd): 2024-06-21
Enter the second date (yyyy-MM-dd): 2023-12-21
Number of days between 2024-06-21 and 2023-12-21 is: -183
```