Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

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.

```
package com.assig.day20;


import java.io.*;

import java.util.*;


public class WordFrequencyCounter {


    public static void main(String[] args) {

        String inputFilePath = "input.txt";

        String outputFilePath = "output.txt";


        // Read the text file and count word frequencies

        Map<String, Integer> wordCountMap =
readFileAndCountWords(inputFilePath);


        // Write the word frequencies to the output file

        writeWordFrequenciesToFile(wordCountMap, outputFilePath);

    }
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
   private static Map<String, Integer> readFileAndCountWords(String
filePath) {
      Map<String, Integer> wordCountMap = new HashMap<>();


      try (FileReader fr = new FileReader(filePath);
          BufferedReader br = new BufferedReader(fr)) {
         String line;
         while ((line = br.readLine()) != null) {
            String[] words = line.split("\\W+");
            for (String word : words) {
               if (!word.isEmpty()) {
                  word = word.toLowerCase();
                  wordCountMap.put(word,
wordCountMap.getOrDefault(word, 0) + 1);
               }
            }
         }
      } catch (IOException e) {
         System.err.println("Error reading file: " + e.getMessage());
      }


      return wordCountMap;
   }
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
   private static void writeWordFrequenciesToFile(Map<String,
Integer> wordCountMap, String filePath) {
      try (FileWriter fw = new FileWriter(filePath);
          BufferedWriter bw = new BufferedWriter(fw)) {
         for (Map.Entry<String, Integer> entry :
wordCountMap.entrySet()) {
            bw.write(entry.getKey() + ": " + entry.getValue());
            bw.newLine();
         }
      } catch (IOException e) {
         System.err.println("Error writing to file: " + e.getMessage());
      }
   }
}
```

Task 2: Serialization and Deserialization

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

```
package com.assig.day20;

import java.io.*;

//Custom class implementing Serializable
class Person implements Serializable {
 private static final long serialVersionUID = 1L;

 private String name;
 private int age;

 public Person(String name, int age) {
     this.name = name;
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

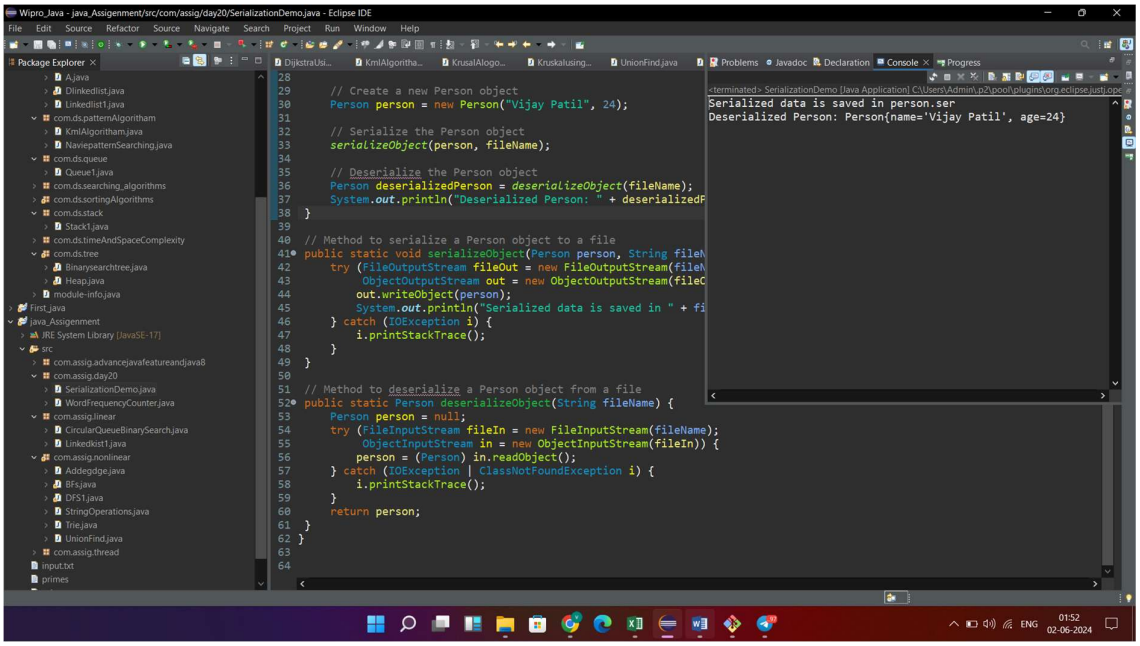
```
        this.age = age;
    }

    @Override
    public String toString() {
        return "Person{name='" + name + "', age=" + age + "}";
    }
}

//Serialization and Deserialization Demo
public class SerializationDemo {

    public static void main(String[] args) {
        String fileName = "person.ser";

        // Create a new Person object
        Person person = new Person("Vijay Patil", 24);

        // Serialize the Person object
        serializeObject(person, fileName);

        // Deserialize the Person object
        Person deserializedPerson = deserializeObject(fileName);
        System.out.println("Deserialized Person: " +
deserializedPerson);
    }

    // Method to serialize a Person object to a file
    public static void serializeObject(Person person, String fileName)
{
        try (FileOutputStream fileOut = new FileOutputStream(fileName);
             ObjectOutputStream out = new ObjectOutputStream(fileOut))
{
            out.writeObject(person);
            System.out.println("Serialized data is saved in " +
fileName);
        } catch (IOException i) {
            i.printStackTrace();
        }
    }

    // Method to deserialize a Person object from a file
    public static Person deserializeObject(String fileName) {
        Person person = null;
        try (FileInputStream fileIn = new FileInputStream(fileName);
             ObjectInputStream in = new ObjectInputStream(fileIn)) {
            person = (Person) in.readObject();
        } catch (IOException | ClassNotFoundException i) {
            i.printStackTrace();
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
        }
        return person;
    }
}
```

Op:



Task 3: New IO (NIO)

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

package com.wipro;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardOpenOption;

import java.util.Iterator;

import java.util.List;

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
public class Mynio {
	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);
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
                        System.out.println("Directory created at " +
cFile.toString());
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        public void readFile() {
            Path f = Paths.get(fileName);
            try {
                List<String> data = Files.readAllLines(f);
                for (String str : data) {
                    System.out.println(str);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }


        }
        public void writeFile(String fileName) {
            Path f = Paths.get(fileName);
            try {
                String content = " Johny Johny , Yes Papa,\n Eating
sugar ? No Papa";
```

```
                Files.write(f, content.getBytes());
                System.out.println("Data Written Successfully");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }

        public void appendFile(String fileName) {
            Path f = Paths.get(fileName);
            try {
                String content = "\n Telling Lies ? No Papa,\n Open
your Mouth, Ha Ha Ha :)";
                Files.write(f, content.getBytes(),
StandardOpenOption.APPEND);
                System.out.println("Data Appended Successfully");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }

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

Name:vijay Patil Email: Vijaypatil454545@gmail.com

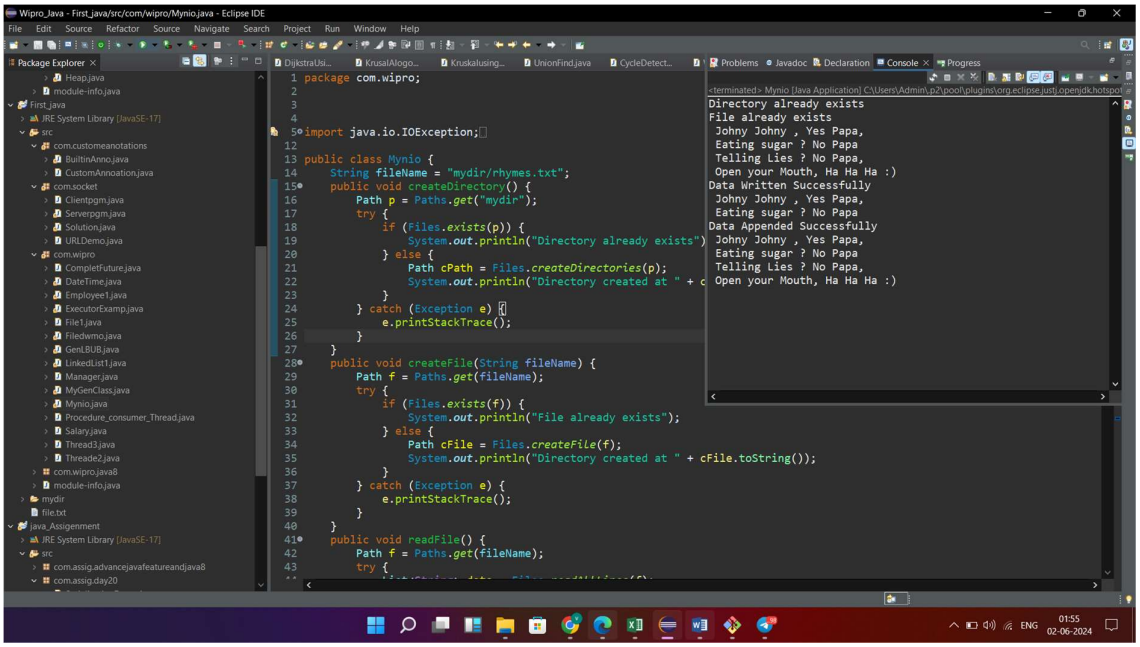
Day 20 assignment

```
            // Create a directory
            mn.createDirectory();

            // Create a file
            mn.createFile("mydir/rhymes.txt");

            // Read from file
            mn.readFile();
            // Write to a file
            mn.writeFile(mn.fileName);

            // Read from file
            mn.readFile();

            // Append to a file
            mn.appendFile(mn.fileName);
            // Read from file
            mn.readFile();
        }

}
```

Op:

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment



Task 4: Java Networking

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

```
package com.socket;


import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.MalformedURLException;

import java.net.URL;

import java.net.URLConnection;


public class URLDemo {
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

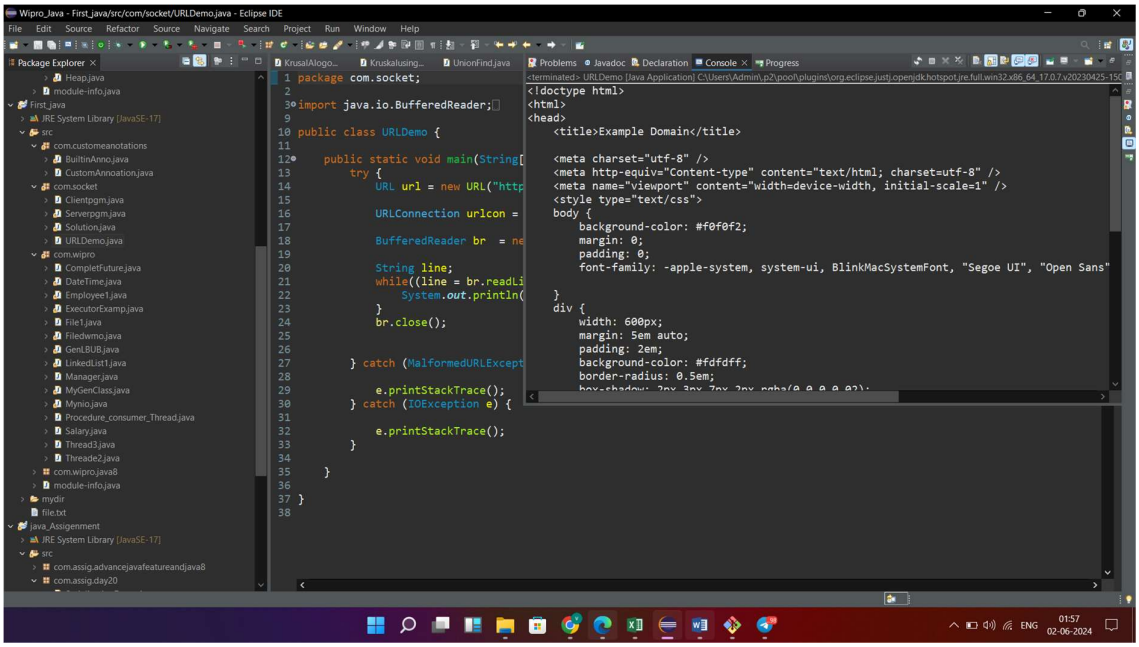
```
public static void main(String[] args) {
        try {
                URL url = new URL("http://www.example.com");

                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();
        }
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
        }


}
```

Op:



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

Operationrequest:

```
package clinentserverapplication;

import java.io.Serializable;

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

    private double number1;
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
    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;
    }
}
```

Opretation server:;

```
package clinentserverapplication;


import java.io.*;
import java.net.*;


public class OperationServer {
   public static void main(String[] args) {
      int port = 12345;


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

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
        while (true) {
            try (Socket socket = serverSocket.accept();
                 ObjectInputStream ois = new
ObjectInputStream(socket.getInputStream());
                 ObjectOutputStream oos = new
ObjectOutputStream(socket.getOutputStream())) {

                OperationRequest request = (OperationRequest)
ois.readObject();
                double result = performOperation(request);

                oos.writeObject(result);
                oos.flush();
            } catch (IOException | ClassNotFoundException ex) {
                ex.printStackTrace();
            }
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
  }

  private static double performOperation(OperationRequest
request) {
    double number1 = request.getNumber1();
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
        double number2 = request.getNumber2();
        String operation = request.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 IllegalArgumentException("Division by zero");
                }
            default:
                throw new UnsupportedOperationException("Unsupported
operation: " + operation);
        }
    }
}
```

Operation client:

```
package clinentserverapplication;
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

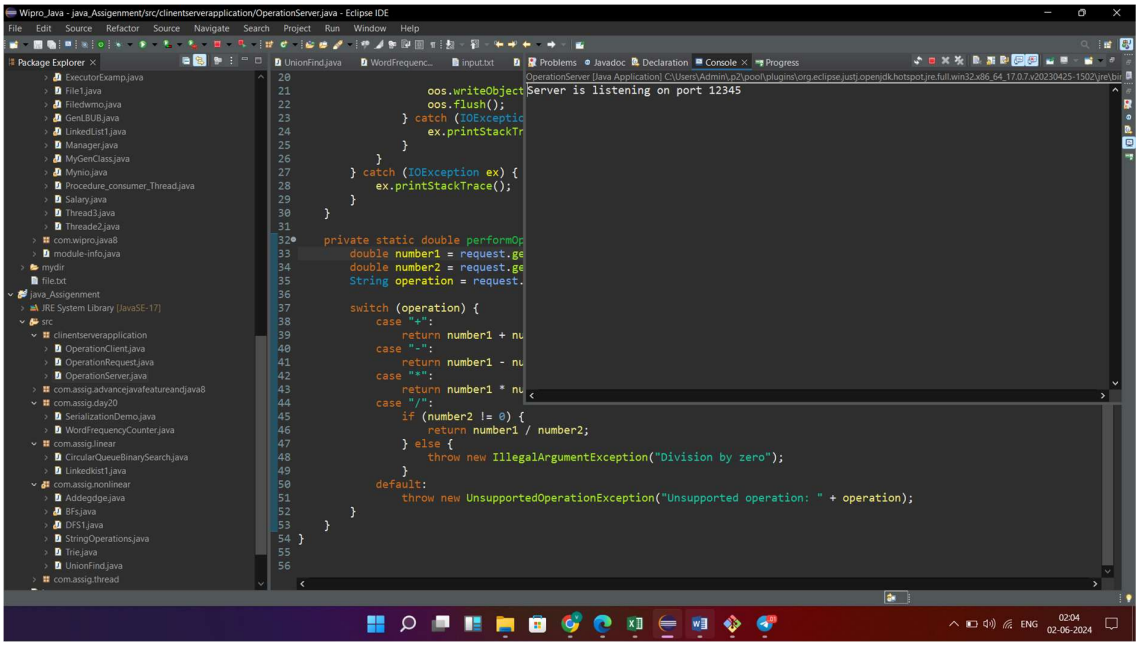
```
import java.io.*;
import java.net.*;

public class OperationClient {
    public static void main(String[] args) {
        String host = "localhost";
        int port = 12345;

        try (Socket socket = new Socket(host, port);
             ObjectOutputStream oos = new
ObjectOutputStream(socket.getOutputStream());
             ObjectInputStream ois = new
ObjectInputStream(socket.getInputStream())) {

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

            double result = (double) ois.readObject();
            System.out.println("Result: " + result);

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

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

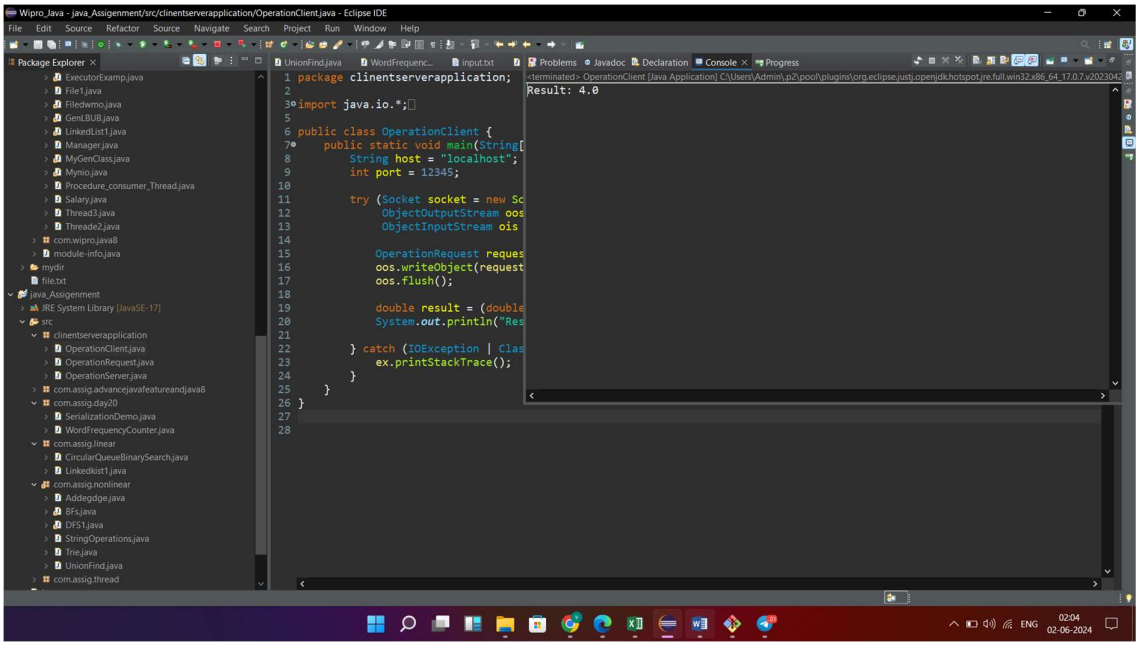
```
        }
    }
}
```

Op:



Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment



Task 6: Java 8 Date and Time API

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

```
package com.assig.day20;


import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.time.temporal.ChronoUnit;

import java.util.Scanner;


public class DateDifferenceCalculator {


    public static void main(String[] args) {
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

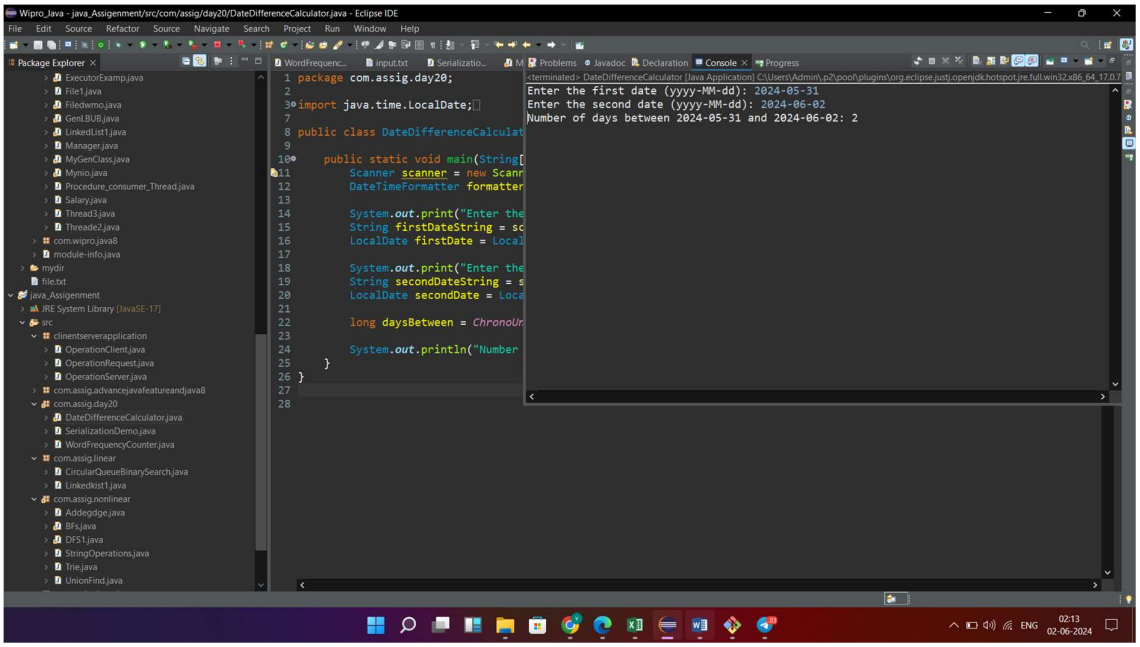
Day 20 assignment

```
        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 + ": " + daysBetween);

    }

}
```

Op:

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment



Task 7: Timezone

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

```
package com.wipro;


import java.time.LocalDate;

import java.time.LocalDateTime;

import java.time.LocalTime;

import java.time.Month;

import java.time.Period;

import java.time.Year;

import java.time.format.DateTimeFormatter;

import java.time.temporal.TemporalAdjuster;

import java.time.temporal.TemporalAdjusters;
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
import java.util.Calendar;
import java.util.TimeZone;

public class DateTime {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		LocalDate localdate =LocalDate.now();
		System.out.println(localdate);

LocalDate cdate=LocalDate.of(2024, Month.MAY, 21);
System.out.println(cdate);
LocalTime tt= LocalTime.now();
System.out.println(tt);

LocalDateTime lt= LocalDateTime.now();
System.out.println(lt);
TimeZone zone = TimeZone.getTimeZone("Asia/Kolkata");
System.out.println("The Offset value of TimeZone: " +
zone.getOffset(Calendar.ZONE_OFFSET));

	Period p=Period.between(localdate, cdate);
	System.out.println(p);
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("HH:mm:ss");
String formattedTime = tt.format(formatter);
System.out.println(formattedTime);



DateTimeFormatter formatter1 =
DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");
String formattedDateTime = lt.format(formatter1);
System.out.println(formattedDateTime);


int year=2024;
System.out.println(Year.isLeap(year));


//TemporalAdjuster class in java
System.out.println("first day of
month"+cdate.with(TemporalAdjusters.firstDayOfMonth()));
System.out.println("first day of next
month"+cdate.with(TemporalAdjusters.firstDayOfNextMonth()));


System.out.println("first day of next
year"+cdate.with(TemporalAdjusters.firstDayOfNextYear()));



 }
```

Name:vijay Patil Email: Vijaypatil454545@gmail.com

Day 20 assignment

```
//      public static void checkdade(LocalDate id)
//      {
//
//              LocalDate today=LocalDate.now();
//              if()
//              System.out.println(id + "is before today");
//      }
```

```
}
```

Op: