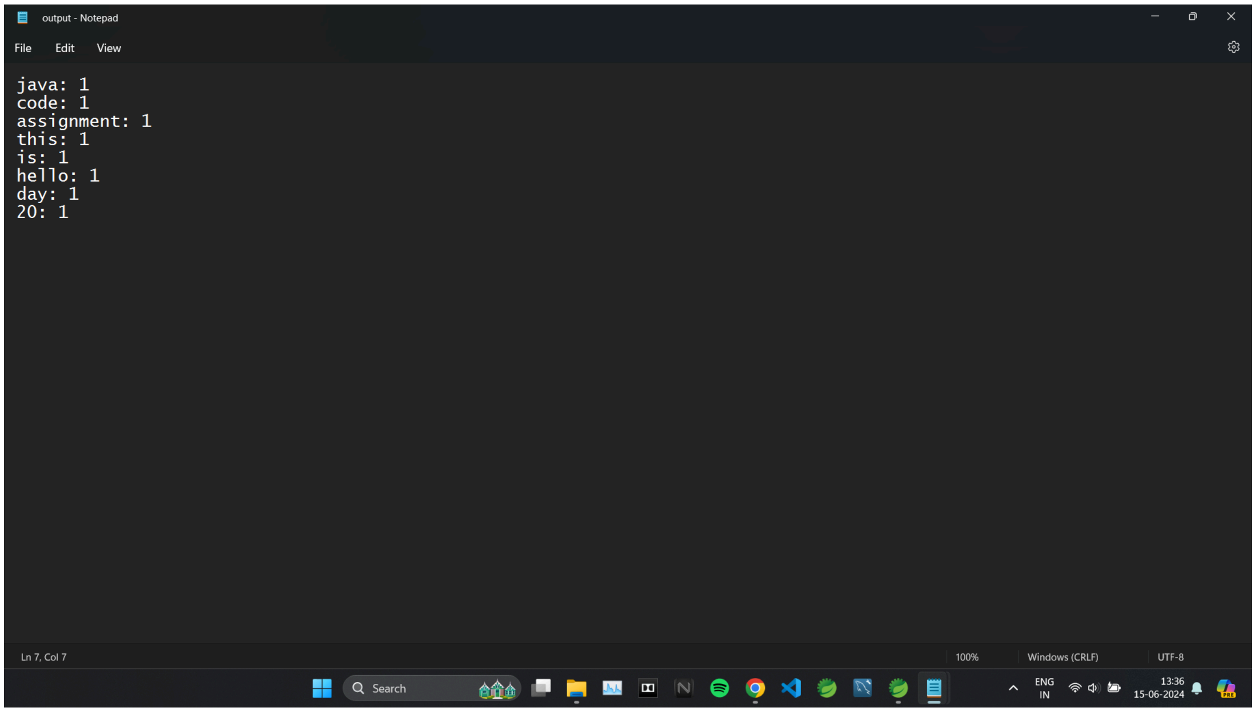
Assignment Day-20
Himanshu Kamane
kamanehimanshu76@gmail.com

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:

```java
package cam.day20;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
public class WordFrequencyCounter {
  public static void main(String[] args) {
      // Change the paths according to your file locations
      String inputFilePath = "input.txt";
      String outputFilePath = "output.txt";
      try {
         Map<String, Integer> wordCounts = countWordFrequencies(inputFilePath);
         writeWordFrequencies(wordCounts, outputFilePath);
      } catch (IOException e) {
         System.err.println("Error: " + e.getMessage());
      }
  }
  private static Map<String, Integer> countWordFrequencies(String filePath) throws
IOException {
      Map<String, Integer> wordCounts = new HashMap<>();
      try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
         String line;
         while ((line = reader.readLine()) != null) {
            String[] words = line.split("\\W+");
            for (String word : words) {
               if (!word.isEmpty()) {
                  word = word.toLowerCase();
                  wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);
               }
            }
        }
    }
    return wordCounts;
}
private static void writeWordFrequencies(Map<String, Integer> wordCounts, String filePath)
throws IOException {
    try (FileWriter writer = new FileWriter(filePath)) {
        for (Map.Entry<String, Integer> entry : wordCounts.entrySet()) {
            writer.write(entry.getKey() + ": " + entry.getValue() + System.lineSeparator());
        }
    }
}
}
```

Output:



```
java: 1
code: 1
assignment: 1
this: 1
is: 1
hello: 1
day: 1
20: 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 cam.day20;
import java.io.*;
```

```java
public class SerializationDemo {
  public static void main(String[] args) {
      Person person = new Person("John Doe", 30);
      String filename = "person.ser";
      // Serialize the Person object
      try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filename)))
{
          oos.writeObject(person);
          System.out.println("Serialization successful: " + person);
      } catch (IOException e) {
          System.err.println("Serialization error: " + e.getMessage());
      }
      // Deserialize the Person object
      try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filename))) {
          Person deserializedPerson = (Person) ois.readObject();
          System.out.println("Deserialization successful: " + deserializedPerson);
      } catch (IOException | ClassNotFoundException e) {
          System.err.println("Deserialization error: " + e.getMessage());
      }
  }
}
```

```java
package cam.day20;
import java.io.Serializable;
public 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;
      this.age = age;
  }
  @Override
  public String toString() {
      return "Person{name='" + name + "', age=" + age + "}";
  }
  // Getters and setters can be added here if needed
}
```

Output:

```
Serialization successful: Person{name='John Doe', age=30}
Deserialization successful: Person{name='John Doe', age=30}
```

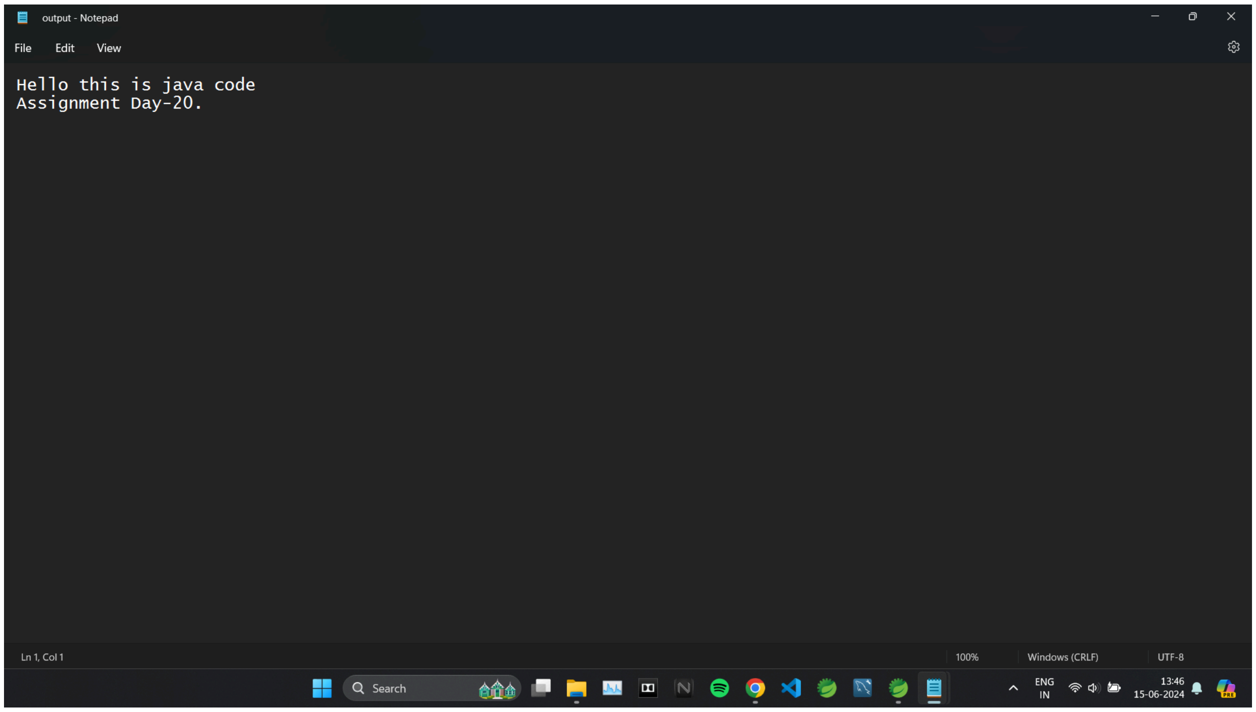
Task 3: New IO (NIO)
Use NIO Channels and Buffers to read content from a file and write to another file.

Solution:

```
package cam.day20;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
public class NIOFileCopy {
  public static void main(String[] args) {
      // Change these paths according to your file locations
      Path inputFilePath = Path.of("input.txt");
      Path outputFilePath = Path.of("output.txt");
      try (
         FileChannel inputChannel = FileChannel.open(inputFilePath,
StandardOpenOption.READ);
         FileChannel outputChannel = FileChannel.open(outputFilePath,
StandardOpenOption.WRITE, StandardOpenOption.CREATE)
      ) {
         ByteBuffer buffer = ByteBuffer.allocate(1024);

         while (inputChannel.read(buffer) > 0) {
            buffer.flip();  // Prepare the buffer for writing
            outputChannel.write(buffer);
            buffer.clear();  // Prepare the buffer for reading
         }
         System.out.println("File copied successfully!");
      } catch (IOException e) {
         System.err.println("I/O error: " + e.getMessage());
      }
  }
}
```

Output:
File copied successfully!


```
Hello this is java code
Assignment Day-20.
```

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 cam.day20;
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;
    }
}
```

```
package cam.day20;
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
public class Server {
    public static void main(String[] args) {
        int port = 12345;
        try (ServerSocket serverSocket = new ServerSocket(port)) {
            System.out.println("Server is listening on port " + port);
            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.getNumber1(), request.getNumber2(),
request.getOperation());
                    oos.writeObject(result);
                } catch (ClassNotFoundException e) {
                    e.printStackTrace();
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```java
  private static double performOperation(double number1, double number2, String
operation) {
      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("Unknown operation: " +
operation);
      }
  }
}
```

```java
package cam.day20;
import java.io.*;
import java.net.Socket;
public class Client {
  public static void main(String[] args) {
      String hostname = "localhost";
      int port = 12345;
      try (Socket socket = new Socket(hostname, port);
          ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());
          ObjectInputStream ois = new ObjectInputStream(socket.getInputStream())) {
         // Example: 2 + 2
         OperationRequest request = new OperationRequest(2, 2, "+");
         oos.writeObject(request);
         double result = (double) ois.readObject();
         System.out.println("Result: " + result);
      } catch (IOException | ClassNotFoundException e) {
         e.printStackTrace();
      }
```

```
    }
}
```

Output:

```
Server is listening on port 12345
```

```
Result: 4.0
```

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 cam.day20;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;
public class DaysBetweenDates {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
        // Input the first date
        System.out.print("Enter the first date (yyyy-MM-dd): ");
        String firstDateStr = scanner.nextLine();
        LocalDate firstDate = LocalDate.parse(firstDateStr, formatter);
        // Input the second date
        System.out.print("Enter the second date (yyyy-MM-dd): ");
        String secondDateStr = scanner.nextLine();
        LocalDate secondDate = LocalDate.parse(secondDateStr, formatter);
        // Calculate the number of days between the two dates
        long daysBetween = ChronoUnit.DAYS.between(firstDate, secondDate);
        // Display the result
        System.out.println("Number of days between " + firstDate + " and " + secondDate + ":
" + daysBetween);
        scanner.close();
    }
}
```

Output:

```
Enter the first date (yyyy-MM-dd): 2023-06-01
Enter the second date (yyyy-MM-dd): 2023-06-15
Number of days between 2023-06-01 and 2023-06-15: 14
```

Task 7: Timezone

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

Solution:

```
package cam.day20;
import java.time.*;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
public class TimezoneConverter {
  public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      // Input the time and timezone
      System.out.print("Enter the time (HH:mm:ss): ");
      String timeStr = scanner.nextLine();
      System.out.print("Enter the timezone of the input time (e.g., Asia/Tokyo): ");
      String inputTimezone = scanner.nextLine();
      System.out.print("Enter the timezone to convert to (e.g., America/New_York): ");
      String outputTimezone = scanner.nextLine();
      // Parse the input time string
      LocalTime time = LocalTime.parse(timeStr,
DateTimeFormatter.ofPattern("HH:mm:ss"));
      // Convert time to ZonedDateTime in the input timezone
      ZonedDateTime inputZonedDateTime = ZonedDateTime.of(LocalDate.now(), time,
ZoneId.of(inputTimezone));
      // Convert to the output timezone
      ZonedDateTime outputZonedDateTime =
inputZonedDateTime.withZoneSameInstant(ZoneId.of(outputTimezone));
      // Format the output time
      LocalTime convertedTime = outputZonedDateTime.toLocalTime();
      String formattedTime =
convertedTime.format(DateTimeFormatter.ofPattern("HH:mm:ss"));
      // Display the converted time
```

```
        System.out.println("Converted time in " + outputTimezone + ": " + formattedTime);
        scanner.close();
    }
}
```

Output:

```
Enter the time (HH:mm:ss): 10:30:00
Enter the timezone of the input time (e.g., Asia/Tokyo): Asia/Tokyo
Enter the timezone to convert to (e.g., America/New_York): America/New_York
Converted time in America/New_York: 21:30:00
```