Day 11 Assignment

1. **Java IO Basics Write a program that reads a text file and counts the frequency of each word using FileReader and FileWriter.**

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

/*
   Java IO Basics Write a program that reads a text file
   and counts the frequency of each word using FileReader and FileWriter.
*/
public class Assignment_1 {
   public static void main(String[] args) {
       Map<String, Integer> mp = new HashMap<>();
       try {
           FileReader file = new
FileReader("C:/Users/coolr/OneDrive/Desktop/Experiments/m5_core_java_programmin
g/day_11/paragraph.txt");
           BufferedReader bf = new BufferedReader(file);
           String line = null;
           while ((line = bf.readLine()) != null) {
               String[] arr = line.split(" ");
               for (String x : arr) {
                   if (mp.containsKey(x)) {
                       mp.put(x, mp.get(x) + 1);
                   } else {
                       mp.put(x, 1);
                   }
               }
           }
       } catch (Exception e) {
           throw new RuntimeException(e);
       }

       System.out.println("Frequency of words in the paragraph.txt :");
       System.out.println(mp);
   }
}
```

Output

```
Frequency of words in the output.txt :
{through=2, butterflies=1, perfect=2, sound=1, flitted=1, trees=1, leaves=1, children=1, adding=1, golden=1, from=1, gentle=1, tall=2, gracefully=1, day=1, they=1, crisp=1,
  rustled=1, joyful=1, soft=1, sky=1, cheerfully=1, vibrant=1, in=2, laughed=1, voices=1, breeze=1, it=1, ancient=1, each=1, echoing=1, morning=1, warm=1, as=1, creating=1,
  glow=1, melodies=1, other=1, soothing=1, ringing=1, color=1, brightly=1, their=2, air=1, birds=1, sun=1, peaceful=1, out=1, The=1, and=1, grass=1, of=2, oak=1, casting=1,
  meadow=1, over=1, a=3, chased=1, green=1, countryside=1, clear=1, was=1, rhythmic=1, splashes=1, played=1, flower=2, the=9, shone=1, blue=1, to=2, serene=2, landscape=1,
 chirped=1}

Process finished with exit code 0
```

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

```
import java.io.*;

class Important implements Serializable {
   private final String importantdata;

   Important(String data) {
       this.importantdata = data;
   }

   @Override
   public String toString() {
       return "Important{" +
               "importantdata='" + importantdata + '\'' +
               '}';
   }
}

public class Assignment_2 {
   public static void main(String[] args) {
       System.out.println("Enter String data to store inside Important
object");
       Important object = new Important("This is a very important message. Save
it in the file");
       try {
           FileOutputStream os = new FileOutputStream("important.txt");
           ObjectOutputStream oos = new ObjectOutputStream(os);

           oos.writeObject(object);

       } catch (IOException e) {
           throw new RuntimeException(e);
       }
       System.out.println("Making object null to mimic data lost");
       object = null;
       System.out.println(object);
       System.out.println("Reading from file to recover the data");
       try {
           FileInputStream fis = new FileInputStream("important.txt");
           ObjectInputStream ois = new ObjectInputStream(fis);
           object = (Important) ois.readObject();
           System.out.println("Object recovered successfully " + object);
       } catch (FileNotFoundException e) {
```

```
            throw new RuntimeException(e);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Output

```
Enter String data to store inside Important object
Making object null to mimic data lost
null
Reading from file to recover the data
Object recovered successfully Important{importantdata='This is a very important message. Save it in the file'}

Process finished with exit code 0
```

---

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

```
import java.io.*;
import java.nio.ByteBuffer;
import java.nio.channels.ReadableByteChannel;
import java.nio.channels.WritableByteChannel;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.nio.file.Path;
import java.nio.file.Paths;

public class Assignment_3 {
    public static void main(String[] args) {
        Path path = Paths.get("m5_core_java_programming/day_11/paragraph.txt");
        try {
            FileInputStream fis = new FileInputStream(path.toFile());
            ReadableByteChannel r = fis.getChannel();
            FileOutputStream fos = new FileOutputStream("newparagraph.txt");
            WritableByteChannel wr = fos.getChannel();
            ByteBuffer buffer = ByteBuffer.allocateDirect(20 * 1024);
            Charset charset = StandardCharsets.UTF_8;
            System.out.println("Reading from paragraph.txt : ");
            while (r.read(buffer) != -1) {
                buffer.flip();
                byte[] bytes = new byte[buffer.remaining()];
                buffer.get(bytes);
                System.out.println(new String(bytes, charset));
                buffer.flip();
                while (buffer.hasRemaining()) {
                    wr.write(buffer);
                }
                buffer.clear();
```

```
            }
            FileReader fr = new FileReader("newparagraph.txt");
            BufferedReader br = new BufferedReader(fr);
            String lines = "";
            System.out.println("Reading from the new txt file newparagraph.txt
:");
            while((lines = br.readLine())!=null){
                System.out.println(lines);
            }
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Output

```
Reading from paragraph.txt :
The sun shone brightly in the clear blue sky casting a warm golden glow over the peaceful
meadow birds chirped cheerfully their melodies echoing through the crisp morning air the
gentle breeze rustled the leaves of the tall ancient oak trees creating a soothing rhythmic
sound butterflies flitted gracefully from flower to flower adding splashes of color to the
vibrant green landscape children laughed and played their joyful voices ringing out as they
chased each other through the tall soft grass it was a perfect perfect day in the serene
serene countryside
Reading from the new txt file newparagraph.txt :
The sun shone brightly in the clear blue sky casting a warm golden glow over the peaceful
meadow birds chirped cheerfully their melodies echoing through the crisp morning air the
gentle breeze rustled the leaves of the tall ancient oak trees creating a soothing rhythmic
sound butterflies flitted gracefully from flower to flower adding splashes of color to the
vibrant green landscape children laughed and played their joyful voices ringing out as they
chased each other through the tall soft grass it was a perfect perfect day in the serene
serene countryside

Process finished with exit code 0
```

---

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

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;

/*
```

```java
   Java Networking Write a simple HTTP client that connects to a URL,
   sends a request, and displays the response headers and body.
*/
public class Assignment_4 {


   public static void main(String []args){
       try {
           URL url = new URL("https://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) {
           throw new RuntimeException(e);
       } catch (IOException e) {
           throw new RuntimeException(e);
       }

   }
}
```

Output

```
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>

Process finished with exit code 0
```

---

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.**

```java
import java.io.Serializable;

public class Num implements Serializable {
    private int num1;
    private int num2;

    public Num(int num1, int num2) {
```

```
        this.num1 = num1;
        this.num2 = num2;
    }

    public int getNum1() {
        return num1;
    }

    public int getNum2() {
        return num2;
    }
}
```

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

/*
   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.
*/
public class Assignment_5_Server {
    static final int port = 2024;

    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(port);
            while (true) {
                Socket socket = ss.accept();
                ObjectInputStream is = new
ObjectInputStream(socket.getInputStream());
                Num num = (Num) is.readObject();
                System.out.println("Received " + num.getNum1() + " and " +
num.getNum2());
                int sum = num.getNum1() + num.getNum2();
                ObjectOutputStream os = new
ObjectOutputStream(socket.getOutputStream());
                System.out.println("Sending back sum to client");
                os.writeObject(sum);
                if (is.readObject() == null) {
                    System.out.println("Closing down Server");
                    break;
                }
            }
        } catch (IOException e) {
```

```
            throw new RuntimeException(e);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
}
```

```
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.Scanner;

public class Assignment_5_Client {
    static final int port = 2024;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        try {
            Socket socket = new Socket("localhost", port);
            System.out.println("Enter two numbers");
            System.out.println("Enter number1");
            int num1 = scan.nextInt();
            System.out.println("Enter number2");
            int num2 = scan.nextInt();
            Num num = new Num(num1, num2);
            ObjectOutputStream oos = new
ObjectOutputStream(socket.getOutputStream());
            System.out.println("Sending Numbers to Server");
            oos.writeObject(num);
            ObjectInputStream is = new
ObjectInputStream(socket.getInputStream());
            System.out.println("Received Sum from the Server");
            System.out.println("The sum is = " + is.readObject());
            oos.writeObject(null);
        } catch (IOException e) {
            throw new RuntimeException(e);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Output

```
m5_core_java_programming.day_11.Assignment_5_Client
Enter two numbers
Enter number1
1234
Enter number2
24323
Sending Numbers to Server
Received Sum from the Server
The sum is = 25557
```

```
m5_core_java_programming.day_11.Assignment_5_Server
Received 1234 and 24323
Sending back sum to client
Closing down Server

Process finished with exit code 0
```

---

6. **Java 8 Date and Time API Write a program that calculates the number of days between two dates input by the user.**

```java
import java.time.LocalDate;
import java.time.Period;
import java.util.Scanner;

public class Assignment_6 {
   public static void main(String[] args) {
        System.out.println("Enter first date :");
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter Day :");
        int day = scan.nextInt();
        System.out.println("Enter Month :");
        int month = scan.nextInt();
        System.out.println("Enter Year :");
        int year = scan.nextInt();

        LocalDate before = LocalDate.of(year, month, day);

        System.out.println("Enter second date :");
        System.out.println("Enter Day :");
        day = scan.nextInt();
        System.out.println("Enter Month :");
        month = scan.nextInt();
        System.out.println("Enter Year :");
```

```
        year = scan.nextInt();

        LocalDate after = LocalDate.of(year, month, day);

        Period period = Period.between(before, after);

        int numOfDays = Math.abs(period.getDays() + period.getMonths() * 30 +
period.getYears() * 365);

        System.out.println("Number of days between " + before + " and " + after
+ " are " + numOfDays);
    }
}
```

Output

```
Enter first date :
Enter Day :
23
Enter Month :
2
Enter Year :
1999
Enter second date :
Enter Day :
26
Enter Month :
5
Enter Year :
2024
Number of days between 1999-02-23 and 2024-05-26 are 9218

Process finished with exit code 0
```

---

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

```
import java.time.*;
import java.util.Scanner;

public class Assignment_7 {
    public static void main(String[] args) {
        ZoneId curr = ZoneId.systemDefault();
        Scanner scan = new Scanner(System.in);
        System.out.println("Current time zone is " + curr);
        System.out.println("Enter your time");
```

```
        System.out.println("Hours");
        int h = scan.nextInt();
        System.out.println("Minutes");
        int m = scan.nextInt();
        System.out.println("Seconds");
        int s = scan.nextInt();
        LocalDateTime time = LocalDateTime.of(LocalDate.now(), LocalTime.of(h,
m, s));

        ZoneId newzone = ZoneId.of("America/New_York");
        ZonedDateTime convertedTime = ZonedDateTime.of(time,
curr).withZoneSameInstant(newzone);

        // Output the converted time
        System.out.println("Time in " + newzone + ": " +
convertedTime.toLocalTime());
    }
}
```

Output

```
Current time zone is Asia/Calcutta
Enter your time
Hours
22
Minutes
52
Seconds
45
Time in America/New_York: 13:22:45

Process finished with exit code 0
```

Tools Used :

IntelliJ IDE
java version "1.8.0_411"
Java(TM) SE Runtime Environment (build 1.8.0_411-b09)
Java HotSpot(TM) Client VM (build 25.411-b09, mixed mode, sharing)
ChatGPT to fill paragraphs in paragraph.txt.