LAB Assignment – 5

Subject : Programming with Java

1. Write a program that takes input from the user, stores it in a file, and then reads the content of that file and prints it to the console.

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

public class FileReadWrite {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter text to store in file:");
        String userInput = scanner.nextLine();

        String fileName = "user_data.txt";

        // Writing to file
        try (FileWriter writer = new FileWriter(fileName)) {
            writer.write(userInput);
            System.out.println("Data written to file successfully.");
        } catch (IOException e) {
            System.out.println("Error writing to file: " + e.getMessage());
        }

        // Reading from file
        System.out.println("Reading data from file:");
        try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("Error reading from file: " + e.getMessage());
        }

        scanner.close();
    }
}

/*
OUTPUT
Data written to file successfully.
Reading data from file:
Hello, this is a Java file handling example.
*/
```

2. Write a java code and use character streams to read data from a .doc file and write data to a text file.

```
import java.io.*;

public class DocToTxtConverter {
    public static void main(String[] args) {
        String inputFile = "input.doc";  // Input .doc file
        String outputFile = "output.txt"; // Output .txt file

        try (FileReader reader = new FileReader(inputFile);
             FileWriter writer = new FileWriter(outputFile)) {

            int character;
            while ((character = reader.read()) != -1) {
                writer.write(character);
            }

            System.out.println("Data successfully copied from " + inputFile +
" to " + outputFile);
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

/*
OUTPUT
Data successfully copied from input.doc to output.txt
*/
```

3. Write a program to perform read and write data using Byte streams.

```
import java.io.*;

public class ByteStreamExample {
    public static void main(String[] args) {
        String inputFile = "input.bin";  // Input file
        String outputFile = "output.bin"; // Output file

        try (FileInputStream inputStream = new FileInputStream(inputFile);
             FileOutputStream outputStream = new FileOutputStream(outputFile))
{

            int byteData;
            while ((byteData = inputStream.read()) != -1) {
                outputStream.write(byteData);
            }

            System.out.println("Data successfully copied from " + inputFile +
" to " + outputFile);
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

/*
OUTPUT
Data successfully copied from input.bin to output.bin
*/
```

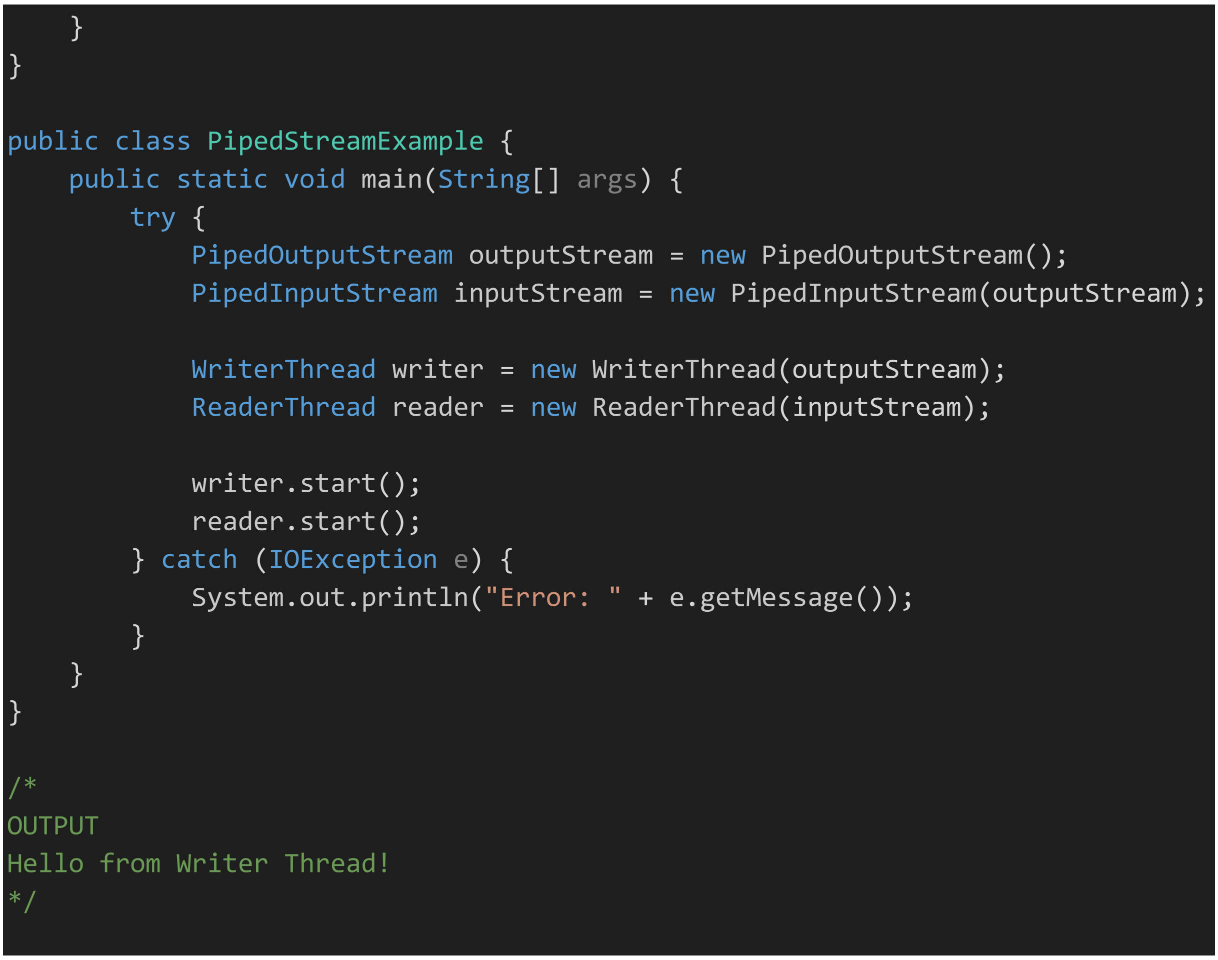
4. Write a program that takes a sentence as input and uses StringTokenizer to split the sentence into words.

```
import java.util.StringTokenizer;
import java.util.Scanner;

public class StringTokenizerExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String sentence = scanner.nextLine();

        // Using StringTokenizer to split sentence into words
        StringTokenizer tokenizer = new StringTokenizer(sentence);

        System.out.println("Words in the sentence:");
        while (tokenizer.hasMoreTokens()) {
            System.out.println(tokenizer.nextToken());
        }

        scanner.close();
    }
}

/*
OUTPUT
Enter a sentence:
Java is a powerful language.

Words in the sentence:
```

```
Java
is
a
powerful
language.
*/
```

5. Write a program to demonstrate PipedStream.

```
import java.io.*;

class WriterThread extends Thread {
    private PipedOutputStream outputStream;

    public WriterThread(PipedOutputStream outputStream) {
        this.outputStream = outputStream;
    }

    public void run() {
        try {
            String message = "Hello from Writer Thread!";
            outputStream.write(message.getBytes());
            outputStream.close();
        } catch (IOException e) {
            System.out.println("Error in WriterThread: " + e.getMessage());
        }
    }
}

class ReaderThread extends Thread {
    private PipedInputStream inputStream;

    public ReaderThread(PipedInputStream inputStream) {
        this.inputStream = inputStream;
    }

    public void run() {
        try {
            int data;
            while ((data = inputStream.read()) != -1) {
                System.out.print((char) data);
            }
            inputStream.close();
        } catch (IOException e) {
            System.out.println("Error in ReaderThread: " + e.getMessage());
        }
```



```
    }
}

public class PipedStreamExample {
    public static void main(String[] args) {
        try {
            PipedOutputStream outputStream = new PipedOutputStream();
            PipedInputStream inputStream = new PipedInputStream(outputStream);

            WriterThread writer = new WriterThread(outputStream);
            ReaderThread reader = new ReaderThread(inputStream);

            writer.start();
            reader.start();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

/*
OUTPUT
Hello from Writer Thread!
*/
```