Reading & Writing File Data

In Java, you can use various classes to read from and write to files. The most commonly used classes for this purpose are `***FileReader***`, `***FileWriter***`, `***BufferedReader***`, and `***BufferedWriter***`.

Below, I'll provide a simple example that demonstrates how to write data to a file and then read it back.

Example : Writing and Reading Data from a File

1. Writing Data to a File

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class FileWrite
{
	public static void main(String[] args)
	{
		String filename = "example.txt";

		// Data to be written to the file
		String[] data = {
					"Hello, World!",
					"Welcome to Java File I/O.",
					"This is a simple example.",
					"Goodbye!"
				 };

		try (BufferedWriter writer = new BufferedWriter(new
		FileWriter(filename)))
		{
			for (String line : data)
			{
				writer.write(line);
				writer.newLine(); // Write a new line after each entry
			}
			System.out.println("Data written to the file successfully.");
		 }
```

```
                catch (IOException e)
                {
                        System.out.println("An error occurred while writing to the file: " +
                        e.getMessage());
                }
        }
}
```

2. Reading Data from a File

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class FileRead
{
        public static void main(String[] args)
        {
                String filename = "example.txt";

                try (BufferedReader reader = new BufferedReader(new
                FileReader(filename)))
                {
                        String line;
                        while ((line = reader.readLine()) != null)
                        {
                                System.out.println(line); // Print each line read from the file
                        }
                }
                catch (IOException e)
                {
                        System.out.println("An error occurred while reading the file: " +
                        e.getMessage());
                }
        }
}
```

Explaination:

1. Writing to a File (*FileWriteExample*):

   - We create a *BufferedWrite* wrapped around a *FileWriter*.

   - The *try-with-resources* statement ensures that the *BufferedWriter* is closed automatically after use.

   - We write several lines of text to a file named example.txt.

2. Reading from a File (*FileReadExample*):

   - We create a *BufferedReader* wrapped around a *FileReader*.

   - Again, we use the *try-with-resources* statement to ensure proper resource management.

   - We read lines from the file in a loop and print each line to the console.

Running the Programs

1. First, run the *FileWriteExample* program to create and write data to example.txt.

2. Then, run the *FileReadExample* program to read and display the content of the file.

Output

After running both programs, the output from the FileReadExample program will be:

```
Hello, World!

Welcome to Java File I/O.

This is a simple example.

Goodbye!
```