WEEK-1 :Design Principles And Patterns

**Exercise 1: Implementing the Singleton Pattern**

**Scenario:** You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Implementation:**

Logger.Java

package singleton;

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger instance created.");

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

        System.out.println("Log: " + message);

    }

}

Main.java

package singleton;

public class Main {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("This is the first log message.");

        Logger logger2 = Logger.getInstance();

        logger2.log("This is the second log message.");

        if (logger1 == logger2) {

            System.out.println("Both logger1 and logger2 are the same instance.");

        } else {

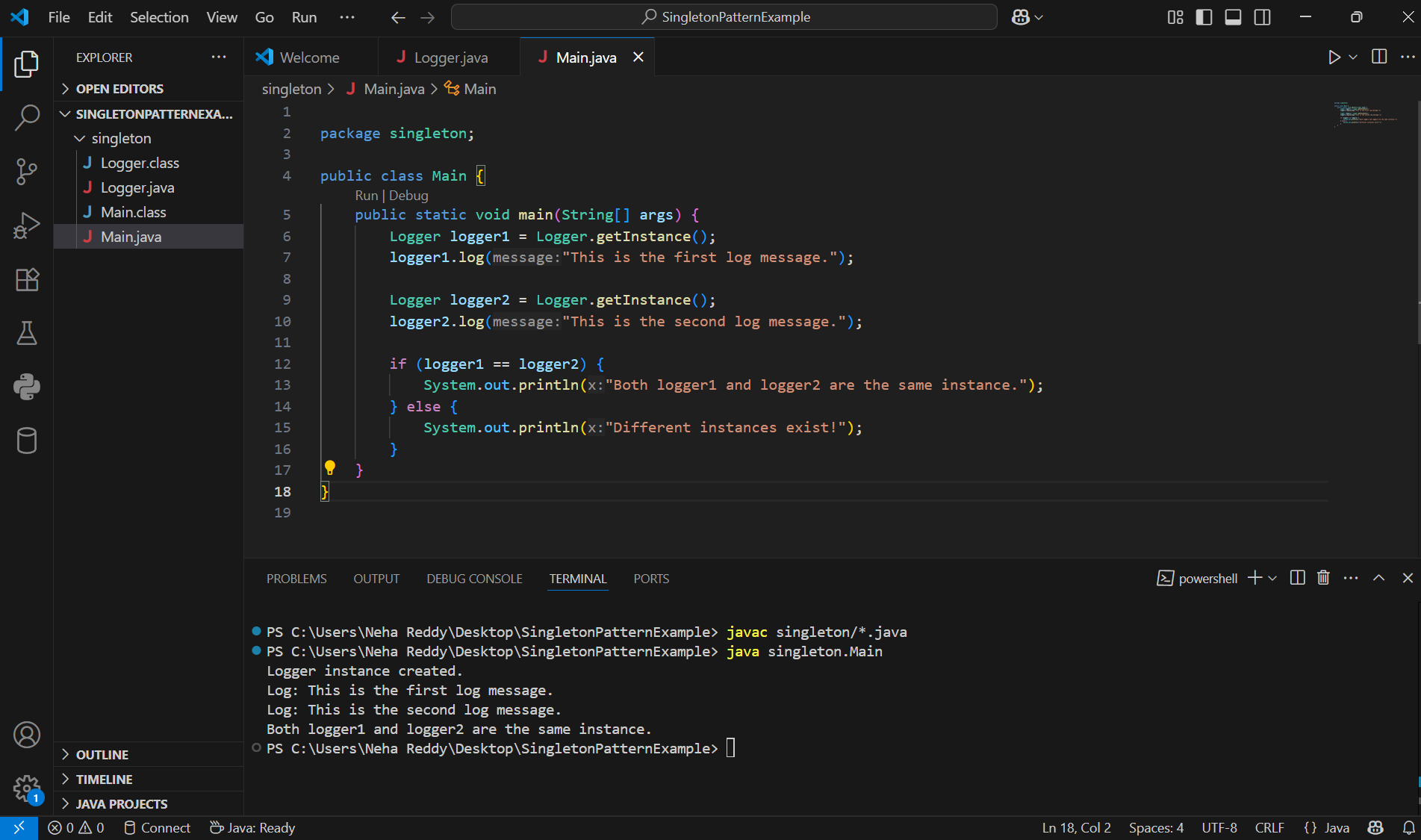
            System.out.println("Different instances exist!");

        }

    }

}

Output:



**Exercise 2: Implementing the Factory Method Pattern**

**Scenario:**You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

Document.java

package documents;

public interface Document {

    void open();

}

WordDocument.java

package documents;

public class WordDocument implements Document {

    public void open() {

        System.out.println("Opening a Word document.");

    }

}

PdfDocument.java

package documents;

public class PdfDocument implements Document {

    public void open() {

        System.out.println("Opening a PDF document.");

    }

}

ExcelDocument.java

package documents;

public class ExcelDocument implements Document {

    public void open() {

        System.out.println("Opening an Excel document.");

    }

}

DocumentFactory.java

package documents;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

WordDocumentFactory.java

package documents;

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

PdfDocumentFactory.java

package documents;

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

ExcelDocumentFactory.java

package documents;

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

Output:

