**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  
  
**Java Code:  
Logger.java :**

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger initialized.");

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

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

    }

}

**LoggerTest.java :**  
  
public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

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

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

        if (logger1 == logger2) {

            System.out.println("Both logger instances are the same (Singleton works).");

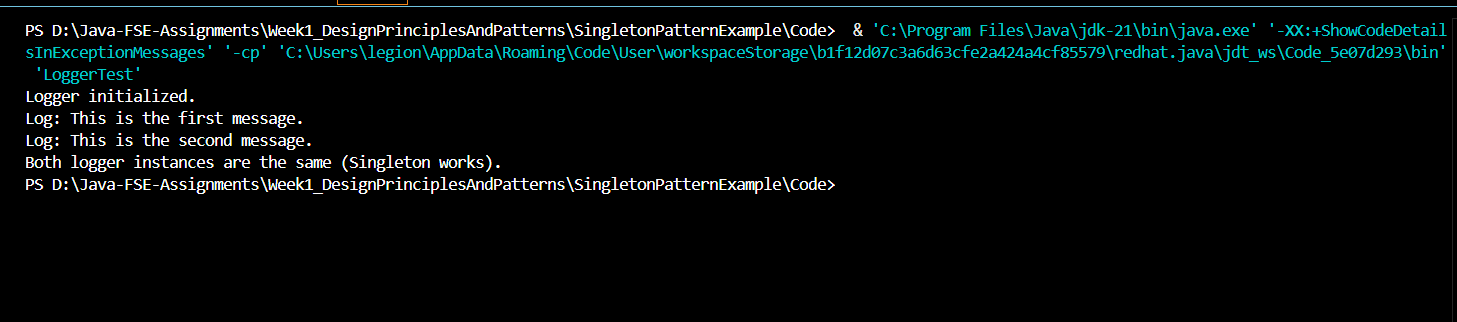
        } else {

            System.out.println("Logger instances are different (Singleton failed).");

        }

    }

}  
  
**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.  
  
**Java Code:**

**FactoryPattern.java :**public class FactoryPattern {

    // Document interface

    interface Document {

        void open();

    }

    // Concrete implementations

    static class WordDocument implements Document {

        public void open() {

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

        }

    }

    static class PdfDocument implements Document {

        public void open() {

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

        }

    }

    static class ExcelDocument implements Document {

        public void open() {

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

        }

    }

    // Factory abstract class

    static abstract class DocumentFactory {

        public abstract Document createDocument();

    }

    // Concrete factories

    static class WordFactory extends DocumentFactory {

        public Document createDocument() {

            return new WordDocument();

        }

    }

    static class PdfFactory extends DocumentFactory {

        public Document createDocument() {

            return new PdfDocument();

        }

    }

    static class ExcelFactory extends DocumentFactory {

        public Document createDocument() {

            return new ExcelDocument();

        }

    }

    // Main method

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordFactory();

        Document word = wordFactory.createDocument();

        word.open();

        DocumentFactory pdfFactory = new PdfFactory();

        Document pdf = pdfFactory.createDocument();

        pdf.open();

        DocumentFactory excelFactory = new ExcelFactory();

        Document excel = excelFactory.createDocument();

        excel.open();

    }

}

**OUTPUT:**