**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

package singleton;

public class Logger {

    private static Logger instance;

    private Logger() {

    }

    public static Logger getInstance() {

        if (instance == null) {

            synchronized (Logger.class) {

                if (instance == null) {

                    instance = new Logger();

                }

            }

        }

        return instance;

    }

    public void log(String message) {

        System.out.println("[LOG] " + message);

    }

}

**LoggerTest.java**

package singleton;

public class LoggerTest {

    public static void main(String[] args) {

        Logger loggerOne = Logger.getInstance();

        Logger loggerTwo = Logger.getInstance();

        loggerOne.log("First message from loggerOne.");

        loggerTwo.log("Second message from loggerTwo.");

        if (loggerOne == loggerTwo) {

            System.out.println("Confirmed: Both are the same instance.");

        } else {

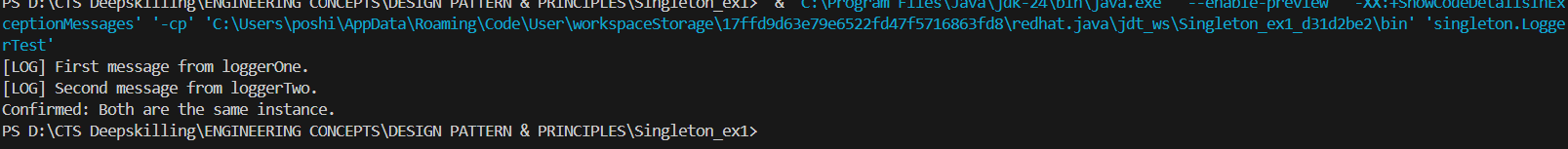
            System.out.println("Error: Different logger instances.");

        }

    }

}

**Output:**

****

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

**Document.java**

package factorymethod;

public interface Document {

    void open();

}

**DocumentFactory.java**

package factorymethod;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelDocument.java**

package factorymethod;

public class ExcelDocument implements Document {

    public void open() {

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

    }

}

**ExcelDocumentFactory.java**

package factorymethod;

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**FactoryMethodTest.java**

package factorymethod;

public class FactoryMethodTest {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word = wordFactory.createDocument();

        word.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf = pdfFactory.createDocument();

        pdf.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel = excelFactory.createDocument();

        excel.open();

    }

}

**PdfDocument.java**

package factorymethod;

public class PdfDocument implements Document {

    public void open() {

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

    }

}

**PdfDocumentFactory.java**

package factorymethod;

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordDocument.java**

package factorymethod;

public class WordDocument implements Document {

    public void open() {

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

    }

}

**WordDocumentFactory.java**

package factorymethod;

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

**Output:**

