**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

public class Logger {

    private static Logger singleLogger;

    private Logger() {

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

    }

    public static Logger getLogger() {

        if (singleLogger == null) {

            singleLogger = new Logger();

        }

        return singleLogger;

    }

    public void write(String message) {

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

    }

}

**Main.java**  
public class Main {

    public static void main(String[] args) {

        Logger firstLogger = Logger.getLogger();

        firstLogger.write("Starting the system");

        Logger secondLogger = Logger.getLogger();

        secondLogger.write("Still using the same logger");

        if (firstLogger == secondLogger) {

            System.out.println("Confirmed: Only one logger exists");

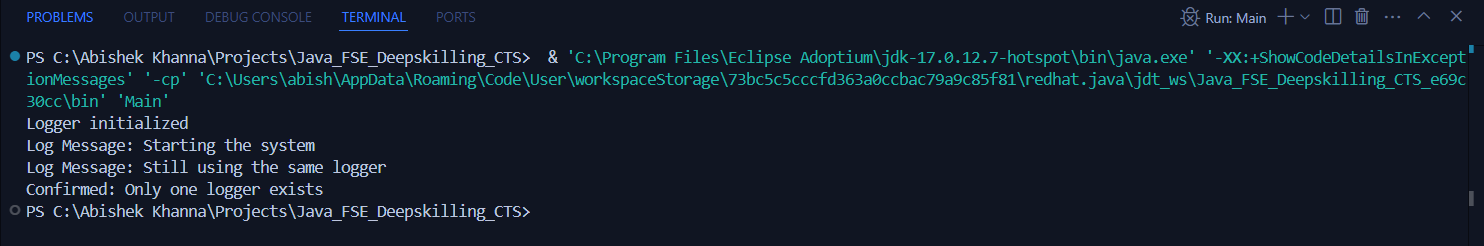
        } else {

            System.out.println("Different loggers exist — something went wrong");

        }

    }

}

**Output :**  


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

**Document.java**

public interface Document {

    void open();

}

**DocumentFactory.java**public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelDocument.java**  
public class ExcelDocument implements Document {

    public void open() {

        System.out.println("Excel document opened.");

    }

}

**ExcelFactory.java**  
public class ExcelFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**Main.java**  
public class Main {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

**PdfDocument.java**

public class PdfDocument implements Document {

    public void open() {

        System.out.println("PDF document opened.");

    }

}

**PdfFactory.java**  
public class PdfFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordDocument.java**

public class WordDocument implements Document {

    public void open() {

        System.out.println("Word document opened.");

    }

}

**WordFactory.java**

public class WordFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

**Output:**  
