**Exercise 1: Implementing the Singleton Pattern**

**Code:**

// package singleton;

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);

    }

}

// package singleton;

public class Main {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

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

        Logger logger2 = Logger.getInstance();

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

        if (logger1 == logger2) {

            System.out.println("Same logger instance used.");

        } else {

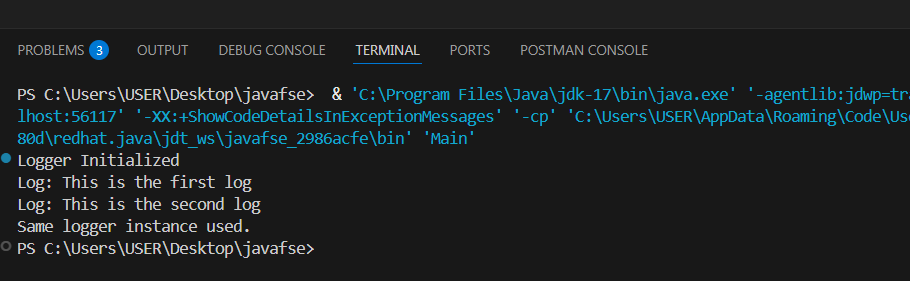
            System.out.println("Different instances found!");

        }

    }

}

**Output:**



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

**Code:**

// package week1handson.factorymethod;

public class Main {

    interface Document {

        void open();

    }

    static class WordDocument implements Document {

        public void open() {

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

        }

    }

    static class PdfDocument implements Document {

        public void open() {

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

        }

    }

    static class ExcelDocument implements Document {

        public void open() {

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

        }

    }

    static abstract class DocumentFactory {

        public abstract Document createDocument();

    }

    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();

        }

    }

    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:**

