**Exercise 1: Implementing the Singleton Pattern**

**CODE FOR LOGGER CLASS**

public class Logger {

    private static Logger instance;

    private Logger() {

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

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();  // Only creates the object if it doesn’t exist

        }

        return instance;

    }

    public void log(String message) {

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

    }

}

**TEST CLASS FOR LOGGERS**

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("Good Morning");

        logger2.log("Hello World");

        if (logger1 == logger2) {

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

        } else {

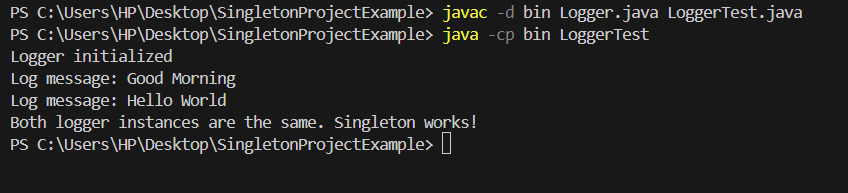
            System.out.println("Different logger instances found. Singleton failed!");

        }

    }

}

OUTPUT



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

* DOCUMENT INTERFACE:

public interface Document {

    void open();

}

* CREATING DOCUMENT CLASSES

public class ExcelDocument implements Document {

    @Override

    public void open() {

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

    }

}

public class PdfDocument implements Document {

    @Override

    public void open() {

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

    }

}

public class WordDocument implements Document {

    @Override

    public void open() {

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

    }

}

* CREATING ABSTRACT CLASS DocumentFactory

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

* CREATING FACTORY CLASSES

public class ExcelDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new ExcelDocument();

    }

}

public class PdfDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new PdfDocument();

    }

}

public class WordDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new WordDocument();

    }

}

* CREATING A TEST CLASS

public class FactoryMethodTest {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

* OUTPUT

