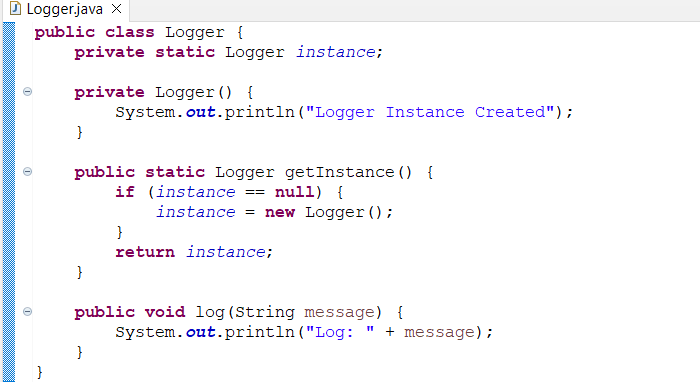
**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

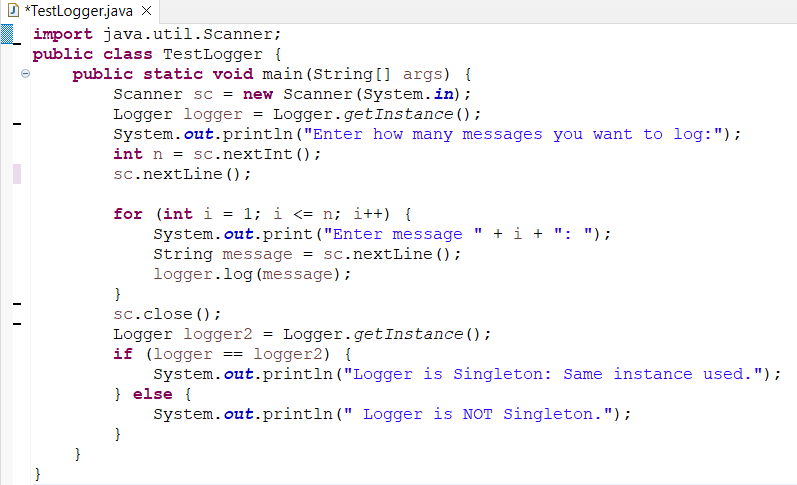
You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Solution:**

**Step 1: create a Logger class**

****

**Step 2: Create a Test Class (TestJogger.java)**

****

**Step 1: create a Logger class**

public class Logger {

private static Logger instance;

private Logger() {

System.out.println("Logger Instance Created");

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

**Step 2: Create a Test Class (TestJogger.java)**

public class TestLogger {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("Logging from logger1");

logger2.log("Logging from logger2");

if (logger1 == logger2) {

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

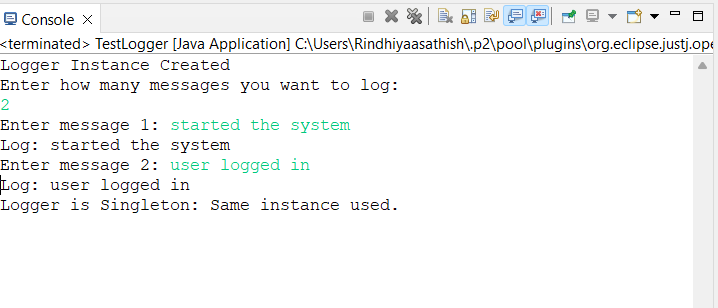
} else {

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

}

}}

**Output**

****

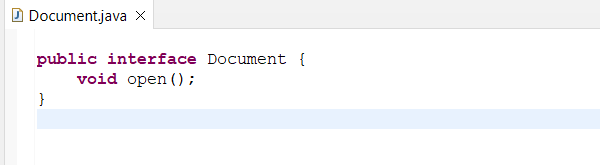
**Exercise 2: Implementing the Factory Method Pattern**

**Scenario:**

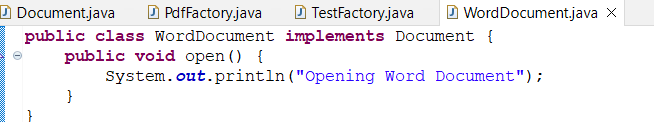
You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Solution:**

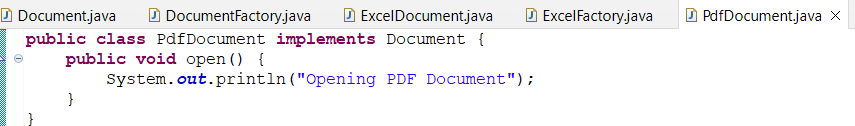
1. **Document.java — Interface**

****

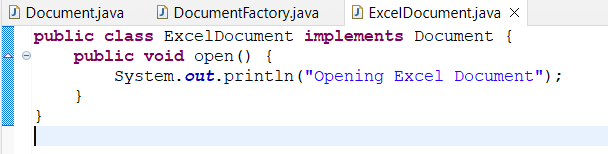
**2.WordDocument.java**

****

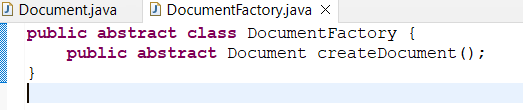
**3.PdfDocument.java**

****

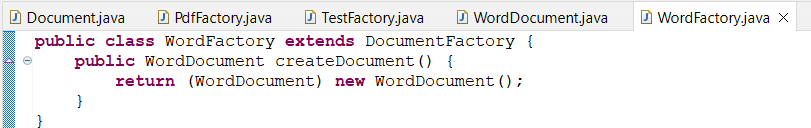
**4.ExcelDocument.java**

****

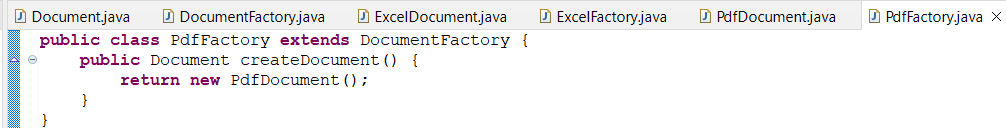
**5. DocumentFactory.java — Abstract Factory**

****

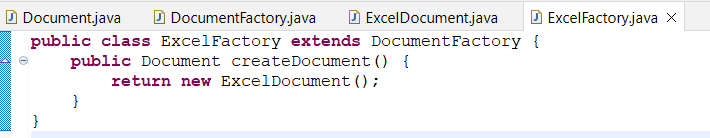
**6.WordFactory.java**

****]

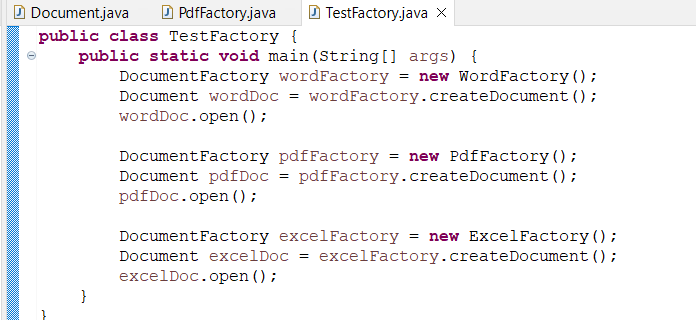
**7.PdfFactory.java**

****

**8.ExcelFactory.java**

****

**9.TestFactory.java — Test Class**



**Code :**

**Document.java**

public interface Document {

void open();

}

**WordDocument.java**

public class WordDocument implements Document {

public void open() {

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

}

}

**PdfDocument.java**

public class PdfDocument implements Document {

public void open() {

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

}

**}**

**ExcelDocument.java**

public class ExcelDocument implements Document {

public void open() {

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

}

}

**DocumentFactory.java**

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**WordFactory.java**

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

**TestFactory.java**

public class TestFactory {

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();

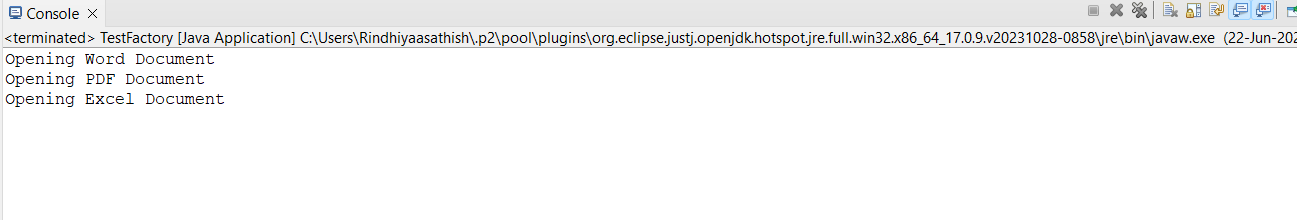
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output**

****