**Week 1\_DesignPrinciplesAndPatterns\_HandsOn**

**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

**package** singleton\_pattern;

**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: " + message);

}

}

**LoggerTest.java**

**package singleton\_pattern;**

public class Loggertest {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("First log message");

Logger logger2 = Logger.*getInstance*();

logger2.log("Second log message");

if (logger1 == logger2) {

System.*out*.println("Both references point to the same Logger instance.");

} else {

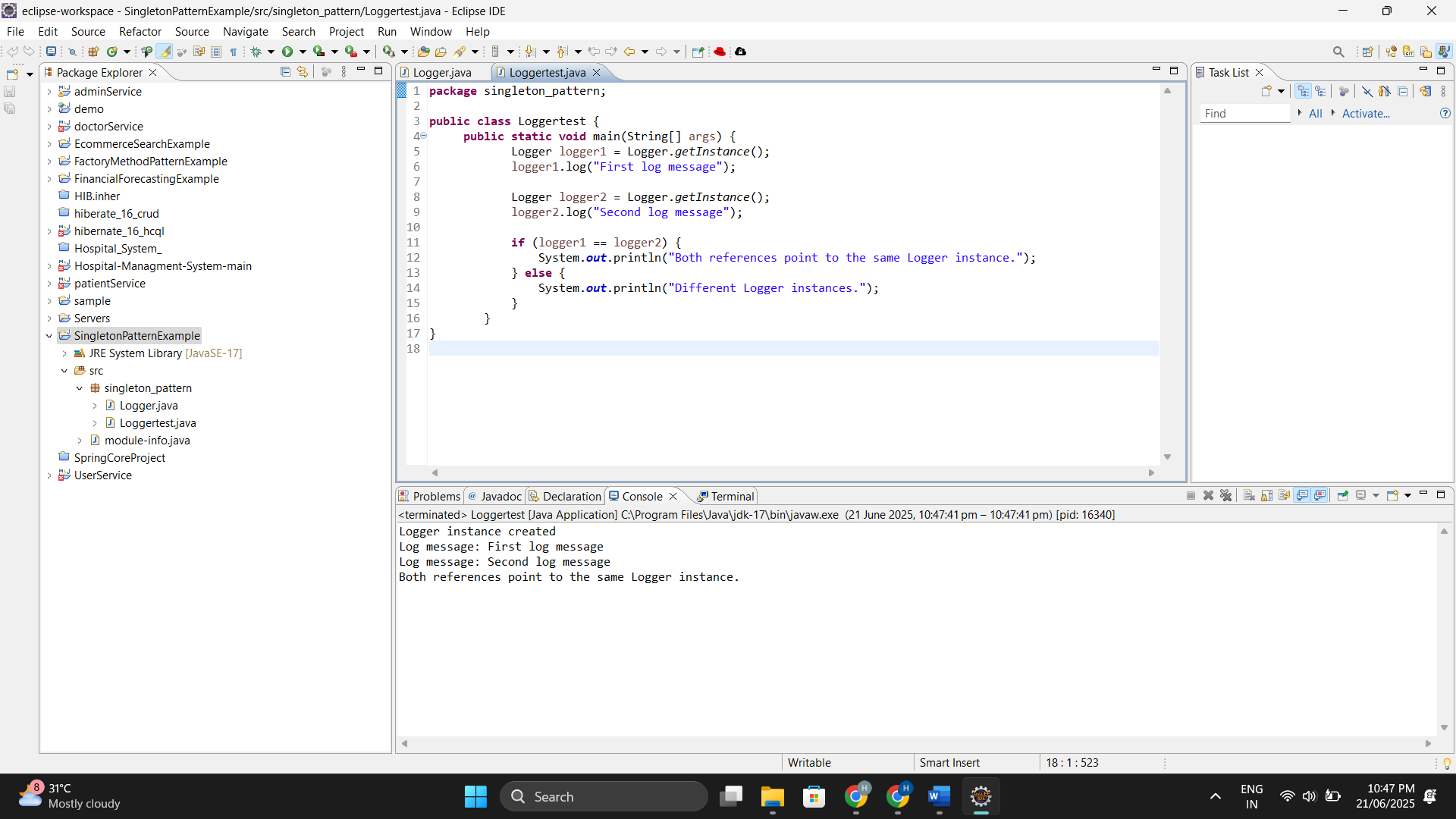
System.*out*.println("Different Logger instances.");

}

}

}

**OUTPUT:**



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

**Document.java**

package documents;

public interface Document {

void open();

}

**ExcelDocument.java**

package documents;

**public** **class** ExcelDocument **implements** Document {

**public** **void** open() {

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

}

}

**PdfDocument.java**

package documents;

public class PdfDocument implements Document {

public void open() {

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

}

}

**WordDocument.java**

package documents;

public class WordDocument implements Document {

public void open() {

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

}

}

**DocumentFactory.java**

package factories;

import documents.Document;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**ExcelDocumentFactory.java**

package factories;

import documents.Document;

import documents.ExcelDocument;

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

**PdfDocumentFactory.java**

package factories;

import documents.Document;

import documents.PdfDocument;

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

**WordDocumentFactory.java**

package factories;

import documents.Document;

import documents.WordDocument;

public class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

**FactoryMethodPatternTest.java**

package test;

import factories.DocumentFactory;

import factories.WordDocumentFactory;

import factories.PdfDocumentFactory;

import factories.ExcelDocumentFactory;

import documents.Document;

public class FactoryMethodPatternTest {

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();

}

}

