**WEEK – 1(Hands-on)**

**Design Principles & Patterns**

**Exercise 1: Implementing the Singleton Pattern**

**Code:**

package com.example.singleton;

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);

}

}

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

if (logger1 == logger2) {

System.*out*.println("Both logger1 and logger2 refer to the same instance.");

} else {

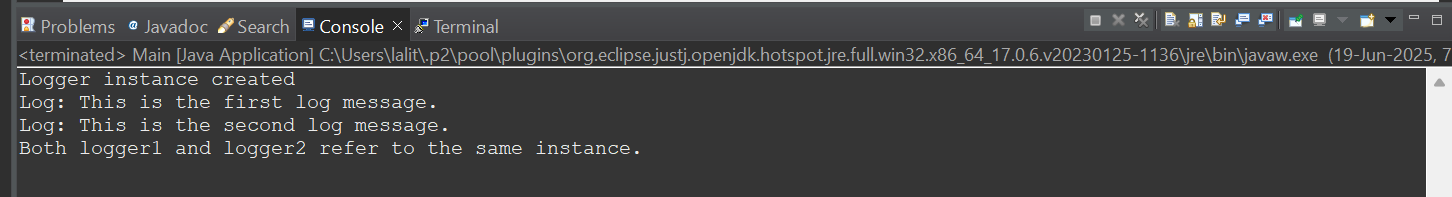
System.*out*.println("Different instances exist! Singleton not implemented correctly.");

}

}

}

**OUTPUT:**

****

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

**Code:**

package com.example.factory;

public interface Document {

void open();

}

public class WordDocument implements Document {

public void open() {

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

}

}

public class PdfDocument implements Document {

public void open() {

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

}

}

public class ExcelDocument implements Document {

public void open() {

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

}

}

public abstract class DocumentFactory {

public abstract Document createDocument();

}

public class WordDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new WordDocument();

}

}

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

public class Main {

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();

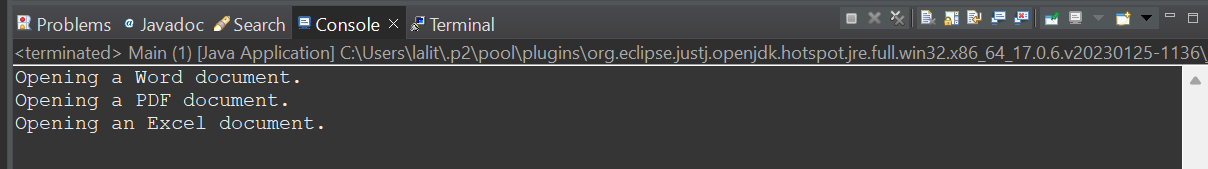
Document excel = excelFactory.createDocument();

excel.open();

}

}

**OUTPUT:**

****