**WEEK – 1 : DESIGN PRINCIPLES AND PATTERN**

**EXERCISE 1 : IMPLEMENTING THE SINGLETON PATTERN**

**Logger.java:**

package singletonpattern;

public class Logger {

private static Logger *instance* = new Logger();

private Logger() {

System.***out***.println("Logger instance created.");

}

public static Logger getInstance() {

return *instance*;

}

public void log(String message) {

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

}

}

**TestLogger.java:**

package singletonpattern;

public class TestLogger {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

logger1.log("First message from logger1");

logger2.log("Second message from logger2");

if (logger1 == logger2) {

System.***out***.println("Both logger instances are the same. Singleton confirmed.");

} else {

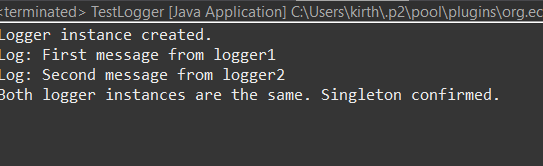
System.***out***.println("Different instances exist. Singleton failed.");

}

}

}

**OUTPUT:**



**EXERCISE 2: IMPLEMENTING THE FACTORY METHOD PATTERN**

**Document.java:**

package com.example.factory;

public interface Document {

void open();

}

**WordDocument.java:**

package com.example.factory;

public class WordDocument implements Document {

*@Override*

public void open() {

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

}

}

**WordDocumentFactory.java:**

package com.example.factory;

public class WordDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new WordDocument();

}

}

**PdfDocument.java:**

package com.example.factory;

public class PdfDocument implements Document {

*@Override*

public void open() {

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

}

}

**PdfDocumentFactory.java:**

package com.example.factory;

public class PdfDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelDocument.java:**

package com.example.factory;

public class ExcelDocument implements Document {

*@Override*

public void open() {

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

}

}

**ExcelDocumentFactory.java:**

package com.example.factory;

public class ExcelDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new ExcelDocument();

}

}

**DocumentFactory.java:**

package com.example.factory;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**DocumentFactoryTest.java:**

package com.example.factory;

public class TestFactoryMethod {

public static void main(String[] args) {

// Word Document Factory

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

// PDF Document Factory

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

// Excel Document Factory

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**OUTPUT:**

