**WEEK – 1**

**Hands-On Tool used**: Eclipse IDE

**Design Principles and Patterns**

**Exercise 1: Implementing the Singleton Pattern**

**Code:**

**Logger.java**

package com.singleton;

public class Logger {

private static Logger instance;

private Logger() {

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

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

**LoggerTest.java**

package com.singleton;

public class LoggerTest {

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.");

}

}

}

**Output:**

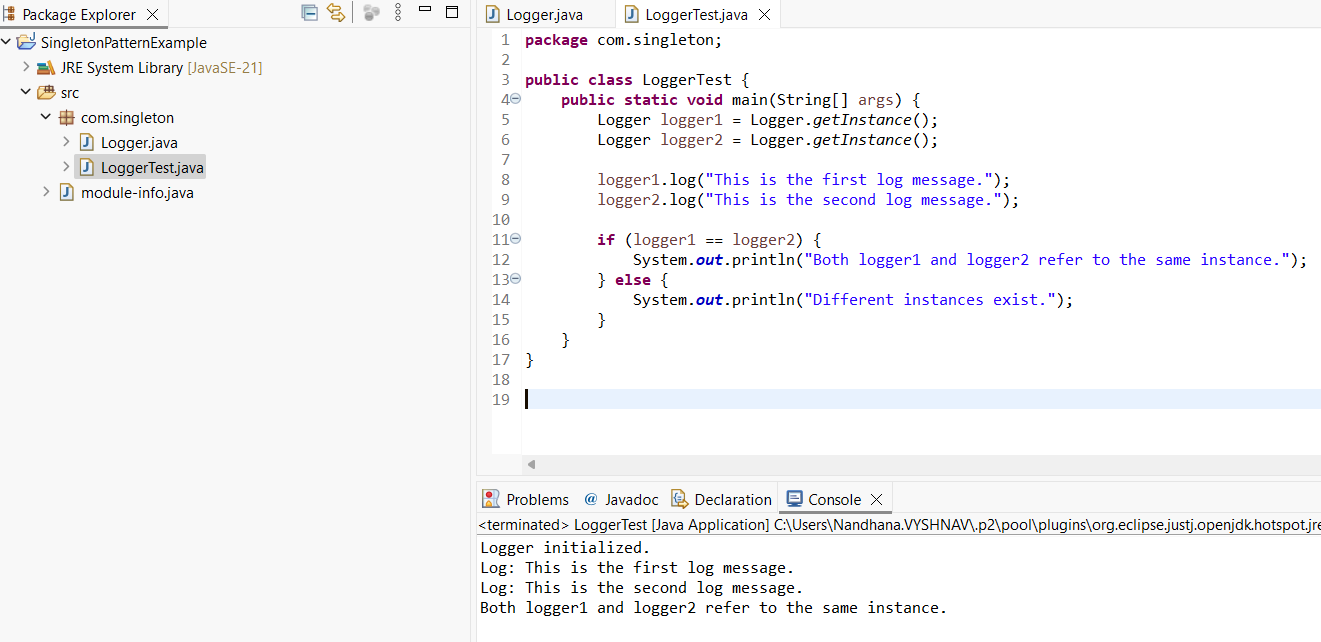


Fig 1.1

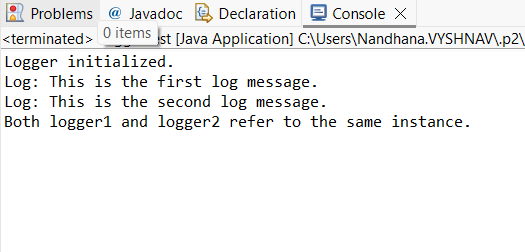


Fig 1.2

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

**Code:**

**Document.java**

package com.factorymethod;

public interface Document {

void open();

}

**WordDocument.java**

package com.factorymethod;

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

**PdfDocument.java**

package com.factorymethod;

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

**ExcelDocument.java**

package com.factorymethod;

public class ExcelDocument implements Document {

@Override

public void open() {

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

}}

**DocumentFactory.java**

package com.factorymethod;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**WordDocumentFactory.java**

package com.factorymethod;

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

**PdfDocumentFactory.java**

package com.factorymethod;

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelDocumentFactory.java**

package com.factorymethod;

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**DocumentTest.java**

package com.factorymethod;

public class DocumentTest {

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();

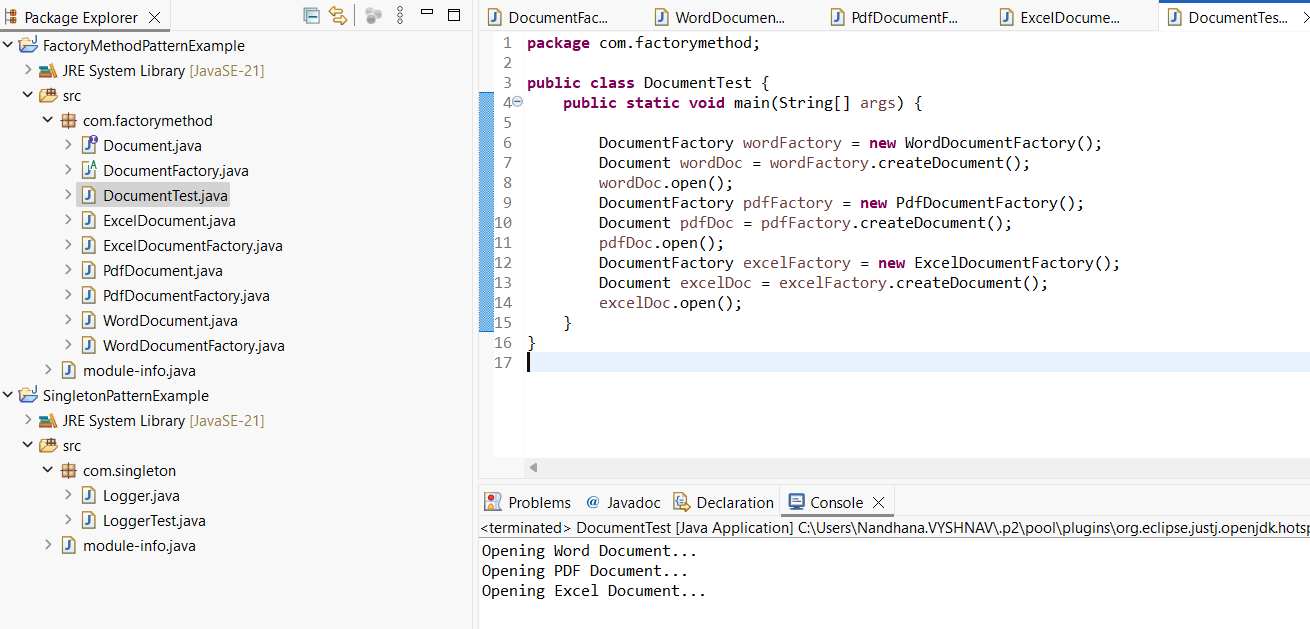
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**

**** Fig 2.1

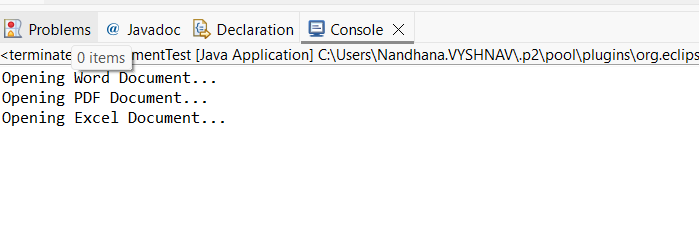
****

Fig 2.2