

Cognizant Digital Nurture 4.0

Name: Siri Chandana Chittipolu

Email: sirichittipolu11@gmail.com

Superset ID: 6386277

Mandatory Hands-On Exercises

Design principles & Patterns

Exercise 1: Implementing the Singleton Pattern

Solution:

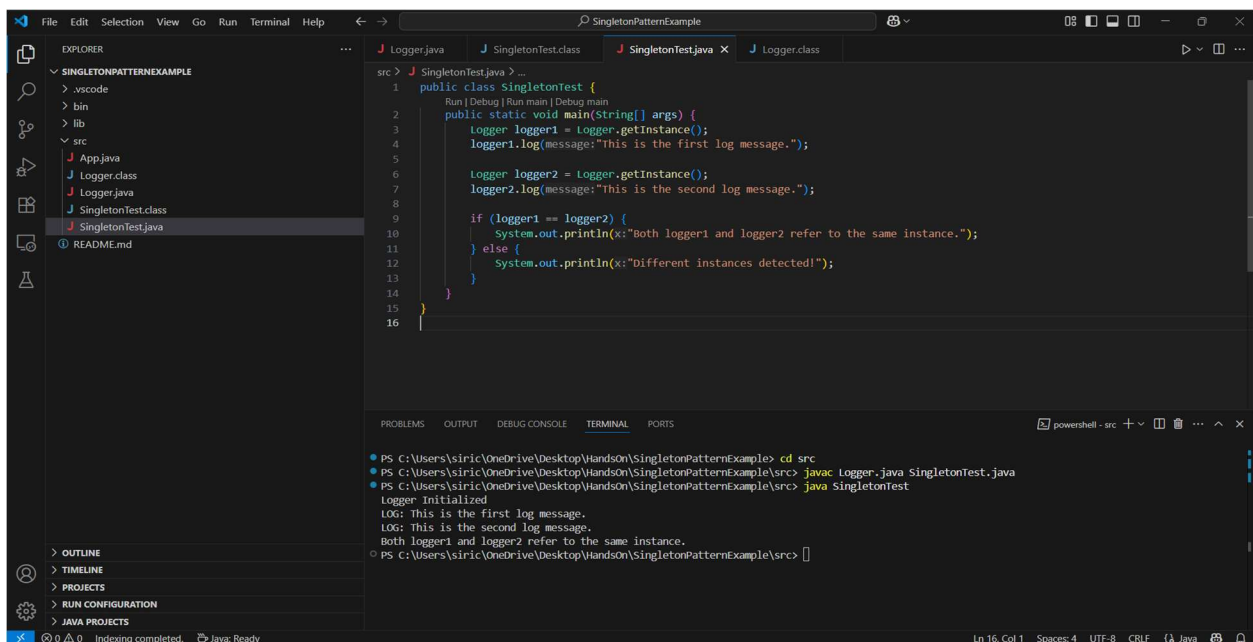
Logger.java:

```
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);  
    }  
}
```

SingletonTest.java :

```
public class SingletonTest {  
    public static void main(String[] args) {  
        Logger logger1 = Logger.getInstance();  
        logger1.log("This is the first log message.");  
        Logger logger2 = Logger.getInstance();  
        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 detected!");  
        }  
    }  
}
```

Output:



The screenshot shows an IDE window titled 'SingletonPatternExample'. The Explorer panel on the left shows a project structure with files like App.java, Logger.class, Logger.java, SingletonTest.class, and SingletonTest.java. The main editor displays the code for SingletonTest.java, which is identical to the code provided in the previous block. The terminal at the bottom shows the following output:

```
PS C:\Users\siric\OneDrive\Desktop\HandsOn\SingletonPatternExample> cd src  
PS C:\Users\siric\OneDrive\Desktop\HandsOn\SingletonPatternExample\src> javac Logger.java SingletonTest.java  
PS C:\Users\siric\OneDrive\Desktop\HandsOn\SingletonPatternExample\src> java SingletonTest  
Logger Initialized  
LOG: This is the first log message.  
LOG: This is the second log message.  
Both logger1 and logger2 refer to the same instance.  
PS C:\Users\siric\OneDrive\Desktop\HandsOn\SingletonPatternExample\src>
```

Exercise 2: Implementing the Factory Method Pattern:

Solution:

Document.java :

```
public interface Document {  
    void open();  
}
```

WordDocument.java :

```
public class WordDocument implements Document {  
    public void open() {  
        System.out.println("Opening a Word document.");  
    }  
}
```

PdfDocument.java :

```
public class PdfDocument implements Document {  
    public void open() {  
        System.out.println("Opening a PDF document.");  
    }  
}
```

ExcelDocument.java :

```
public class ExcelDocument implements Document {  
    public void open() {  
        System.out.println("Opening an Excel document.");  
    }  
}
```

WordDocumentFactory.java :

```
public class WordDocumentFactory extends DocumentFactory {  
    public Document createDocument() {  
        return new WordDocument();  
    }  
}
```

PdfDocumentFactory.java :

```
public class PdfDocumentFactory extends DocumentFactory {  
    public Document createDocument() {  
        return new PdfDocument();  
    }  
}
```

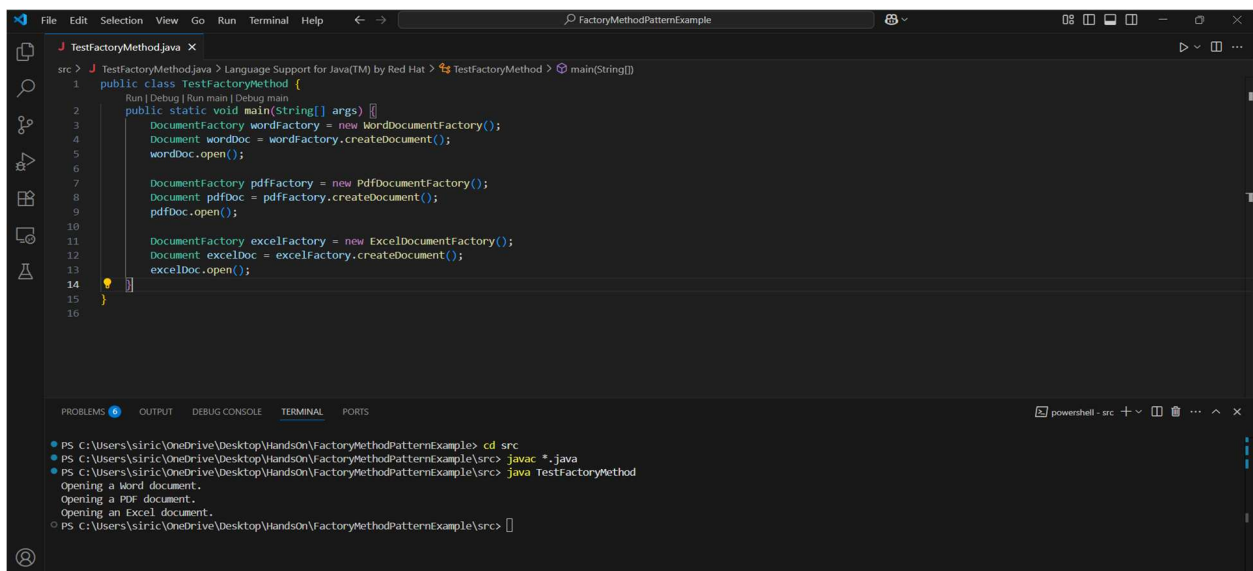
ExceldocumentFactory.java :

```
public class ExcelDocumentFactory extends DocumentFactory {  
    public Document createDocument() {  
        return new ExcelDocument();  
    }  
}
```

TestFactoryMethod.java :

```
public class TestFactoryMethod {  
    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:



The screenshot shows an IDE window titled 'FactoryMethodPatternExample'. The editor displays the 'TestFactoryMethod.java' file with the following code:

```
src > J TestFactoryMethod.java > Language Support for Java(TM) by Red Hat > TestFactoryMethod > main(String[])  
1 public class TestFactoryMethod {  
2     public static void main(String[] args) {  
3         DocumentFactory wordFactory = new WordDocumentFactory();  
4         Document wordDoc = wordFactory.createDocument();  
5         wordDoc.open();  
6  
7         DocumentFactory pdfFactory = new PdfDocumentFactory();  
8         Document pdfDoc = pdfFactory.createDocument();  
9         pdfDoc.open();  
10  
11         DocumentFactory excelFactory = new ExcelDocumentFactory();  
12         Document excelDoc = excelFactory.createDocument();  
13         excelDoc.open();  
14     }  
15 }  
16
```

The bottom panel shows the 'TERMINAL' output:

```
PS C:\Users\siric\OneDrive\Desktop\Wandson\FactoryMethodPatternExample> cd src  
PS C:\Users\siric\OneDrive\Desktop\Wandson\FactoryMethodPatternExample\src> javac *.java  
PS C:\Users\siric\OneDrive\Desktop\Wandson\FactoryMethodPatternExample\src> java TestFactoryMethod  
Opening a Word document.  
Opening a PDF document.  
Opening an Excel document.  
PS C:\Users\siric\OneDrive\Desktop\Wandson\FactoryMethodPatternExample\src>
```