**WEEK-1**

**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 1: Implementing the Singleton Pattern**

**Code :**

**Package name: com.Singleton;**

**File name:Logger.java**

package com.singleton;

class Logger {

/\*private static Logger obj = new Logger(); Eagerly instantiation(This will create the object at the time of class loading and even if we dont use the object,

it will consume lot of system resources)\*/

private static Logger obj;

private Logger() {

}

public static Logger getInstance(){

if (obj==null)

obj = new Logger(); // Lazy instantiation (creating the object only after the object is checked if it is null or not once the static method is called)

return obj;

}

}

**File name:SingletonExample.java**

package com.Singleton;

class SingletonExample {

public static void main(String args[]) {

Logger log1 = Logger.getInstance();

System.out.println("Logger1: "+log1);

Logger log2 = Logger.getInstance();

System.out.println("Logger2: "+log2);

if(log1==log2)

System.out.println("both log1 and log2 are referring to the same instance");

else

System.out.println("both are referring to two different instances");

}

}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

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

**Code:**

**File name- Document.java**

interface Document {

void create();

}

**File name- PdfDocument.java**

public class PdfDocument implements Document {

public void create () {

System.out.println("pdf Document created");

}

}

**File name- WordDocument.java**

public class WordDocument implements Document {

public void create () {

System.out.println("word Document created");

}

}

**File name- ExcelDocument.java**

public class ExcelDocument implements Document {

public void create () {

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

}

}

**File name- DocumentFactory.java**

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**File name- PdfDocumentFactory.java**

public class PdfDocumentFactory extends DocumentFactory

{

public Document createDocument() {

return new PdfDocument();

}

}

**File name- WordDocumentFactory.java**

public class WordDocumentFactory extends DocumentFactory

{

public Document createDocument() {

return new WordDocument();

}

}

**File name-ExcelDocumentFactory.java**

public class ExcelDocumentFactory extends DocumentFactory

{

public Document createDocument() {

return new ExcelDocument();

}

}

**File name-FactoryMethod.java**

public class FactoryMethod {

public static void main(String args[]) {

DocumentFactory pdf = new PdfDocumentFactory();

Document doc1 = pdf.createDocument();

doc1.create();

DocumentFactory word = new WordDocumentFactory();

Document doc2 = word.createDocument();

doc2.create();

DocumentFactory excel = new ExcelDocumentFactory();

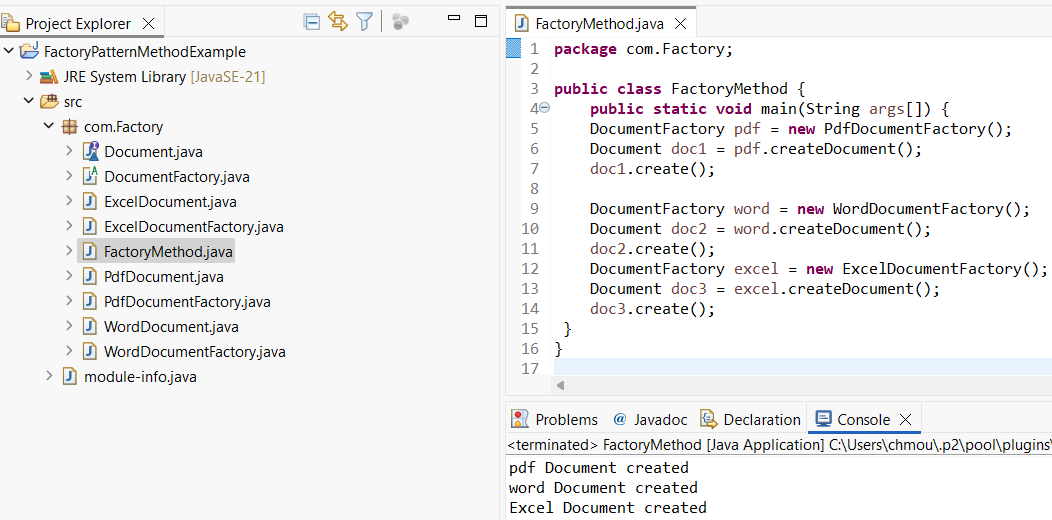
Document doc3 = excel.createDocument();

doc3.create();

}

}

**Output:**

****