**Week 1**

**Design Patterns and Principles**

**Exercise 1:** **Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**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);

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

if (logger1 == logger2) {

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

} else {

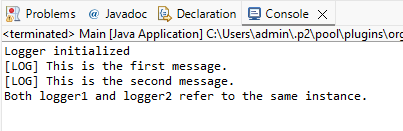
System.*out*.println("Different instances were created (Singleton failed).");

}

}

}

**Output:**

****

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

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Document.java**

**package** com.factory.document;

**public** **interface** Document {

**void** open();

}

**WordDocument.java**

**package** com.factory.document;

**public** **class** WordDocument **implements** Document {

**public** **void** open() {

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

}

}

**PdfDocument.java**

**package** com.factory.document;

**public** **class** PdfDocument **implements** Document {

**public** **void** open() {

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

}

}

**ExcelDocument.java**

**package** com.factory.document;

**public** **class** ExcelDocument **implements** Document {

**public** **void** open() {

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

}

}

**DocumentFactory.java**

**package** com.factory.document;

**public** **abstract** **class** DocumentFactory {

**public** **abstract** Document createDocument();

}

**WordDocumentFactory.java**

**package** com.factory.document;

**public** **class** WordDocumentFactory **extends** DocumentFactory {

**public** Document createDocument() {

**return** **new** WordDocument();

}

}

**PdfDocumentFactory.java**

**package** com.factory.document;

**public** **class** PdfDocumentFactory **extends** DocumentFactory {

**public** Document createDocument() {

**return** **new** PdfDocument();

}

}

**ExcelDocumentFactory.java**

**package** com.factory.document;

**public** **class** ExcelDocumentFactory **extends** DocumentFactory {

**public** Document createDocument() {

**return** **new** ExcelDocument();

}

}

**Main.java**

**package** com.factory.document;

**public** **class** Main {

**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();

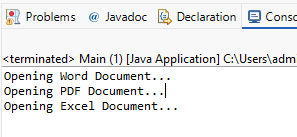
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**

****