***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.**

class Logger {

private static Logger instance = null;

private Logger() {

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

}

public static Logger getLogger() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String msg) {

System.out.println("LOG >> " + msg);

}

}

public class SingletonPatternExample {

public static void main(String[] args) {

Logger logA = Logger.getLogger();

logA.log("System booted");

Logger logB = Logger.getLogger();

logB.log("User signed in");

if (logA == logB) {

System.out.println("Same Logger instance reused");

} else {

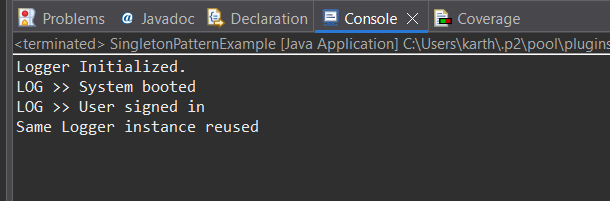
System.out.println(" Different instances created");

}

}

}

***Output:***

******

***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.**

public class DocumentFactoryDemo {

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();

}

}

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

System.out.println("Opening a .docx Word Document with Microsoft Word.");

}

}

class PdfDocument implements Document {

public void open() {

System.out.println("Opening a .pdf Document in Adobe Reader.");

}

}

class ExcelDocument implements Document {

public void open() {

System.out.println("Opening a .xlsx Excel Spreadsheet in Excel.");

}

}

abstract class DocumentFactory {

public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

class ExcelDocumentFactory extends DocumentFactory {

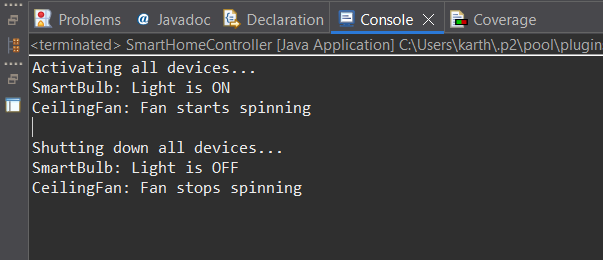
public Document createDocument() {

return new ExcelDocument();

}

}

***Output:***

******