**Module 1: DESIGN PRINCIPLES AND PATTERNS**

**Exercise 1:Implementing the single Pattern**

**CODE :**

public class SingletonExample {

static class Logger {

private static Logger instance;

private Logger() {

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

}

public static synchronized Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

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 logger instances are the same (Singleton confirmed).");

} else {

System.out.println("Different logger instances exist (Singleton violated).");

}

}

}

**OUTPUT:**

A screen shot of a computer

AI-generated content may be incorrect.

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

**CODE:**

public class Main { interface Document {

void open();

void save();

void close();

}

static class WordDocument implements Document {

public void open() { System.out.println("Opening Word Document"); }

public void save() { System.out.println("Saving Word Document"); }

public void close() { System.out.println("Closing Word Document"); }

}

static class PdfDocument implements Document {

public void open() { System.out.println("Opening PDF Document"); }

public void save() { System.out.println("Saving PDF Document"); }

public void close() { System.out.println("Closing PDF Document"); }

}

static class ExcelDocument implements Document {

public void open() { System.out.println("Opening Excel Document"); }

public void save() { System.out.println("Saving Excel Document"); }

public void close() { System.out.println("Closing Excel Document"); }

}

static abstract class DocumentFactory {

public abstract Document createDocument();

public void processDocument() {

Document doc = createDocument();

doc.open();

doc.save();

doc.close();

}

}

static class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

static class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

static class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

public static void main(String[] args) {

DocumentFactory factory;

factory = new WordDocumentFactory();

factory.processDocument();

System.out.println();

factory = new PdfDocumentFactory();

factory.processDocument();

System.out.println();

factory = new ExcelDocumentFactory();

factory.processDocument();

}}

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.