# Week1 - Design Patterns & Principles - HandsOn

## Exercise: SingletonPattern

SingletonPattern.java

class Logger {  
 private static Logger instance;  
  
 private Logger() {}  
  
 public static Logger getInstance() {  
 if (instance == null) {  
 instance = new Logger();  
 }  
 return instance;  
 }  
  
 public void log(String message) {  
 System.out.println("Log: " + message);  
 }  
}  
  
public class SingletonPattern {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.getInstance();  
 Logger logger2 = Logger.getInstance();  
 logger1.log("This is the first log message.");  
 logger2.log("This is the second log message.");  
 System.out.println("Same instance: " + (logger1 == logger2));  
 }  
}

### Output:

A screenshot of a computer

AI-generated content may be incorrect.

## Exercise: FactoryPattern

FactoryPattern.java

interface Document {  
 void open();  
}  
  
class WordDocument implements Document {  
 public void open() {  
 System.out.println("Opening Word document...");  
 }  
}  
  
class PdfDocument implements Document {  
 public void open() {  
 System.out.println("Opening PDF document...");  
 }  
}  
  
class ExcelDocument implements Document {  
 public void open() {  
 System.out.println("Opening Excel document...");  
 }  
}  
  
abstract class DocumentFactory {  
 abstract Document createDocument();  
}  
  
class WordFactory extends DocumentFactory {  
 Document createDocument() {  
 return new WordDocument();  
 }  
}  
  
class PdfFactory extends DocumentFactory {  
 Document createDocument() {  
 return new PdfDocument();  
 }  
}  
  
class ExcelFactory extends DocumentFactory {  
 Document createDocument() {  
 return new ExcelDocument();  
 }  
}  
  
public class FactoryPattern {  
 public static void main(String[] args) {  
 DocumentFactory factory = new PdfFactory();  
 Document doc = factory.createDocument(); // only createDocument() used.  
 doc.open();  
 }  
}

### Output:

A screenshot of a computer

AI-generated content may be incorrect.