**Design Patterns and Principles (Mandatory Hands On)**

**Exercise 1: Implementing the Singleton Pattern**

Code:

Logger.java

*public* class Logger {

*private* *static* Logger instance**;**

*private* Logger() {

        System**.***out***.**println("Logger instance created.")**;**

    }

*public* *static* Logger getInstance() {

**if** (instance **==** null) {

            instance **=** **new** Logger()**;**

        }

**return** instance**;**

    }

*public* *void* log(String **message**) {

        System**.***out***.**println("LOG: " **+** message)**;**

    }

}

TestLogger.java

*public* class TestLogger {

*public* *static* *void* main(String[] **args**) {

        Logger logger1 **=** Logger**.**getInstance()**;**

        Logger logger2 **=** Logger**.**getInstance()**;**

        logger1**.**log("First log message.")**;**

        logger2**.**log("Second log message.")**;**

**if** (logger1 **==** logger2) {

            System**.***out***.**println("Both logger instances are the same.")**;**

        } **else** {

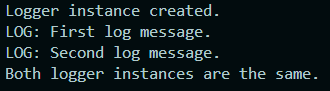
            System**.***out***.**println("Different logger instances exist.")**;**

        }

    }

}

Output:



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

Document.java

*public* interface Document {

*void* open()**;**

}

DocumentFactory.java

*public* *abstract* class DocumentFactory {

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

}

ExcelDocument.java

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

*public* *void* open() {

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

    }

}

ExcelFactory.java

*public* class ExcelFactory *extends* *DocumentFactory* {

*public* Document createDocument() {

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

    }

}

PdfDocument.java

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

*public* *void* open() {

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

    }

}

PdfFactory.java

*public* class PdfFactory *extends* *DocumentFactory* {

*public* Document createDocument() {

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

    }

}

WordDocument.java

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

*public* *void* open() {

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

    }

}

WordFactory.java

*public* class WordFactory *extends* *DocumentFactory* {

*public* Document createDocument() {

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

    }

}

TestFactory.java

*public* class TestFactory {

*public* *static* *void* main(String[] **args**) {

        DocumentFactory wordFactory **=** **new** WordFactory()**;**

        Document word **=** wordFactory**.**createDocument()**;**

        word**.**open()**;**

        DocumentFactory pdfFactory **=** **new** PdfFactory()**;**

        Document pdf **=** pdfFactory**.**createDocument()**;**

        pdf**.**open()**;**

        DocumentFactory excelFactory **=** **new** ExcelFactory()**;**

        Document excel **=** excelFactory**.**createDocument()**;**

        excel**.**open()**;**

    }

}

Output:

