

Exercise 1: Implementing the Singleton Pattern

```
package com.example;

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);
    }
}

package com.example;

public class Main {

    public static void main(String[] args) {
        Logger logger1 = Logger.getInstance();
        logger1.log("First message");
        Logger logger2 = Logger.getInstance();
        logger2.log("Second message");
        if (logger1 == logger2) {
            System.out.println("Both logger instances are the same.");
        } else {
            System.out.println("Different instances exist.");
        }
    }
}
```

```
}  
}
```

Output

```
C:\Users\Admin\Desktop\sagar>javac -d out *.java  
  
C:\Users\Admin\Desktop\sagar>java -cp out com.example.Main  
Logger initialized.  
LOG: First message  
LOG: Second message  
Both logger instances are the same.  
  
C:\Users\Admin\Desktop\sagar>
```

Exercise 2: Implementing the Factory Method Pattern

```
public class TestFactoryPattern {  
  
    static abstract class Document {  
  
        public abstract void open();  
  
    }  
  
    static class WordDocument extends Document {  
  
        public void open() {  
  
            System.out.println("Opening Word Document...");  
  
        }  
  
    }  
  
    static class PdfDocument extends Document {  
  
        public void open() {  
  
            System.out.println("Opening PDF Document...");  
  
        }  
  
    }  
  
    static class ExcelDocument extends Document {  
  
        public void open() {  
  
            System.out.println("Opening Excel Document...");  
  
        }  
  
    }  
  
    static abstract class DocumentFactory {
```

```

    public abstract Document createDocument();
}

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();
    Document doc1 = factory.createDocument();
    doc1.open();

    factory = new PdfDocumentFactory();
    Document doc2 = factory.createDocument();
    doc2.open();

    factory = new ExcelDocumentFactory();

```

```
        Document doc3 = factory.createDocument();  
        doc3.open();  
    }  
}
```

Output

```
C:\Users\Admin\Desktop\sagar>javac TestFactoryPattern.java  
  
C:\Users\Admin\Desktop\sagar>java TestFactoryPattern  
Opening Word Document...  
Opening PDF Document...  
Opening Excel Document...  
  
C:\Users\Admin\Desktop\sagar>
```