

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.

Steps:

Singleton class : Logger.java

```
public class Logger {  
    private static Logger logger;  
    private Logger() {  
    }  
    public static Logger getInstance() {  
        if (logger == null) {  
            logger = new Logger();  
        }  
        return logger;  
    }  
}
```

Implementation in main class : Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Logger obj1 = Logger.getInstance();  
        Logger obj2 = Logger.getInstance();  
  
        System.out.println(obj1 == obj2); // Should print true  
    }  
}
```

Output : True

Exercise 2 : Implementing the Factory Method Pattern

Document Interface : Document.java

```
interface Document {  
    void open();  
}
```

Excel Document class : ExcelDocument.java

```
public class ExcelDocument implements Document {  
    public void open() {  
        System.out.println("Excel document opened");  
    }  
}
```

PDF Document class : PdfDocument.java

```
public class PdfDocument implements Document {  
    public void open() {  
        System.out.println("PDF document opened");  
    }  
}
```

Word Document class : WordDocument.java

```
public class WordDocument implements Document {  
    public void open() {  
        System.out.println("Word document opened");  
    }  
}
```

Abstract class DocumentFactory : DocumentFactory.java

```
abstract class DocumentFactory {  
    abstract public Document createDocument(String str);  
}
```

DocumentFactoryClass class: DocumentFactoryClass.java

```
public class DocumentFactoryClass extends DocumentFactory {  
    public Document createDocument(String str) {  
        if (str.equalsIgnoreCase("ExcelDocument")) {  
            return new ExcelDocument();  
        }  
        if (str.equalsIgnoreCase("PdfDocument")) {  
            return new PdfDocument();  
        }  
        if (str.equalsIgnoreCase("WordDocument")) {  
            return new WordDocument();  
        }  
        return null;  
    }  
}
```

Main class : Main.java

```
public class Main {  
    public static void main(String[] args) {  
        DocumentFactoryClass obj = new DocumentFactoryClass();  
        Document word = obj.createDocument("worddocument");  
        word.open();  
    }  
}
```

Output :

word document opened