\

WEEK 1 : Design Pattern and Principles

Code:

Exercise 1: Implementing the Singleton Pattern

```
public class Main {
    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.");

        if (logger1 == logger2) {
            System.out.println("Both logger instances are the same (Singleton verified).");
        } else {
            System.out.println("Different instances exist (Singleton failed).");
        }
    }
}

class Logger {
    private static final Logger instance = new Logger();

    private Logger() {
```

```
        System.out.println("Logger instance created");
    }

    public static Logger getInstance() {
        return instance;
    }

    public void log(String message) {
        System.out.println("[LOG] " + message);
    }
}
```

Output :

```
Logger instance created
[LOG] This is the first log message.
[LOG] This is the second log message.
Both logger instances are the same (Singleton verified).
```

Code :

Exercise 2: Implementing the Factory Method Pattern

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      DocumentService service = new DocumentService();
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter document type (word, pdf, excel):");
      String type = scanner.nextLine().trim().toLowerCase();
      Document doc = service.createDocument(type);
      if (doc != null) {
         doc.open();
      } else {
         System.out.println("Unsupported document type.");
      }
      scanner.close();
```

```
    }
}

interface Document {
    void open();
}

class WordDocument implements Document {
    public void open() {
        System.out.println("Opening a Word document with formatting support...");
    }
}

class PdfDocument implements Document {
    public void open() {
        System.out.println("Opening a PDF document with secure viewer...");
    }
}

class ExcelDocument implements Document {
    public void open() {
        System.out.println("Opening an Excel spreadsheet with formulas...");
    }
}

interface DocumentFactory {
```

```
    Document createDocument();
}

class WordDocumentFactory implements DocumentFactory {
    public Document createDocument() {
        return new WordDocument();
    }
}

class PdfDocumentFactory implements DocumentFactory {
    public Document createDocument() {
        return new PdfDocument();
    }
}

class ExcelDocumentFactory implements DocumentFactory {
    public Document createDocument() {
        return new ExcelDocument();
    }
}

class DocumentService {
    private final Map<String, DocumentFactory> registry = new HashMap<>();

    public DocumentService() {
        registry.put("word", new WordDocumentFactory());
```

```
        registry.put("pdf", new PdfDocumentFactory());
        registry.put("excel", new ExcelDocumentFactory());
    }

    public Document createDocument(String type) {
        DocumentFactory factory = registry.get(type);
        return factory != null ? factory.createDocument() : null;
    }
}
```

Output :

```
Enter document type (word, pdf, excel):
word
Opening a Word document with formatting support...

Enter document type (word, pdf, excel):
pdf
Opening a PDF document with secure viewer...

Enter document type (word, pdf, excel):
excel
Opening an Excel spreadsheet with formulas...
```