DESIGN PRINCIPLES AND PATTERNS MANDATORY

Superset ID : 6384831

Name : Mohana Priya N

E-mail : mohanapriya.2205056@srec.ac.in

1) Exercise 1: Implementing the Singleton Pattern

Solution:

```
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);
    }
}
public class Main {
    public static void main(String[] args) {
        Logger logger1 = Logger.getInstance();
        Logger logger2 = Logger.getInstance();
        logger1.log("First log message");
        logger2.log("Second log message");
        System.out.println("Same instance? " + (logger1 == logger2));
    }}
```

2) Exercise 2: Implementing the Factory Method Pattern

```
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();
```

```
    }
}
public class TestFactory {
    public static void main(String[] args) {
        DocumentFactory factory = new PdfFactory();
        Document doc = factory.createDocument();
        doc.open();
    }
}
```