WEEK:1

**DESIGN PATTERN AND PRINCIPLES**

# Exercise 1: Implementing the Singleton Pattern

**Code:**

class Logger {

// Step 1: Create a private static instance of Logger private static Logger singleInstance;

// Step 2: Private constructor to prevent instantiation private Logger() {

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

}

// Step 3: Public method to provide access to the instance public static Logger getInstance() {

if (singleInstance == null) { singleInstance = new Logger();

}

return singleInstance;

}

// Logging method

public void log(String message) {

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

}

}

public class Main {

public static void main(String[] args) {

// Get logger instance and log messages Logger logger1 = Logger.getInstance(); logger1.log("This is the first log message.");

Logger logger2 = Logger.getInstance(); logger2.log("This is the second log message.");

// Check if both logger references point to the same object if (logger1 == logger2) {

System.out.println("Both logger instances are the same. Singleton works!");

} else {

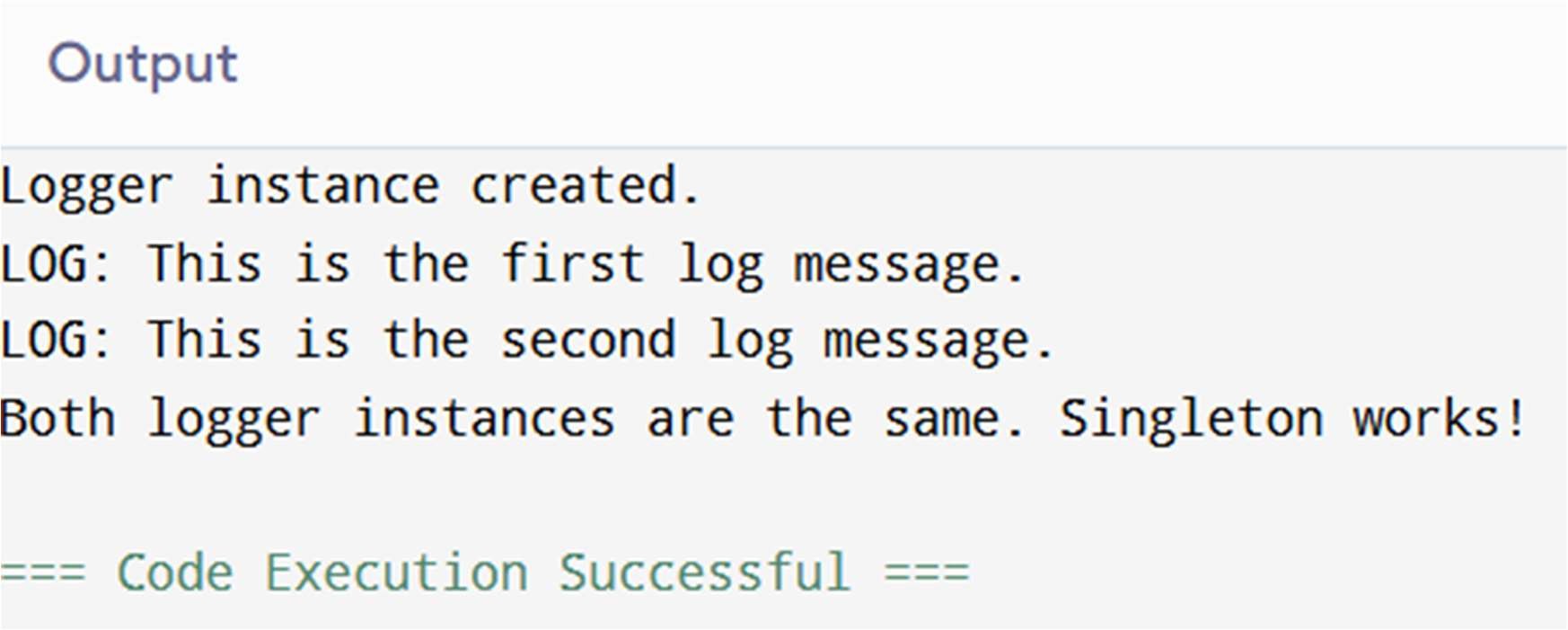
System.out.println("Different logger instances. Singleton failed.");

}

}

}

**Output:**

****

# Exercise 2: Implementing the Factory Method Pattern

**Code:**

interface Document { void open();

void close(); // New method added

}

class WordDocument implements Document { public void open() {

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

}

public void close() {

System.out.println("Closing Word document...");

}

}

class PdfDocument implements Document { public void open() {

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

}

public void close() {

System.out.println("Closing PDF document...");

}

}

class ExcelDocument implements Document { public void open() {

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

}

public void close() {

System.out.println("Closing Excel document...");

}

}

// Abstract Factory

abstract class DocumentFactory {

public abstract Document createDocument();

}

// Concrete Factories

class WordFactory extends DocumentFactory { public Document createDocument() {

return new WordDocument();

}

}

class PdfFactory extends DocumentFactory { public Document createDocument() {

return new PdfDocument();

}

}

class ExcelFactory extends DocumentFactory { public Document createDocument() {

return new ExcelDocument();

}

}

public class Main {

public static void main(String[] args) { DocumentFactory wordFactory = new WordFactory();

Document wordDoc = wordFactory.createDocument(); wordDoc.open();

wordDoc.close();

DocumentFactory pdfFactory = new PdfFactory(); Document pdfDoc = pdfFactory.createDocument(); pdfDoc.open();

pdfDoc.close();

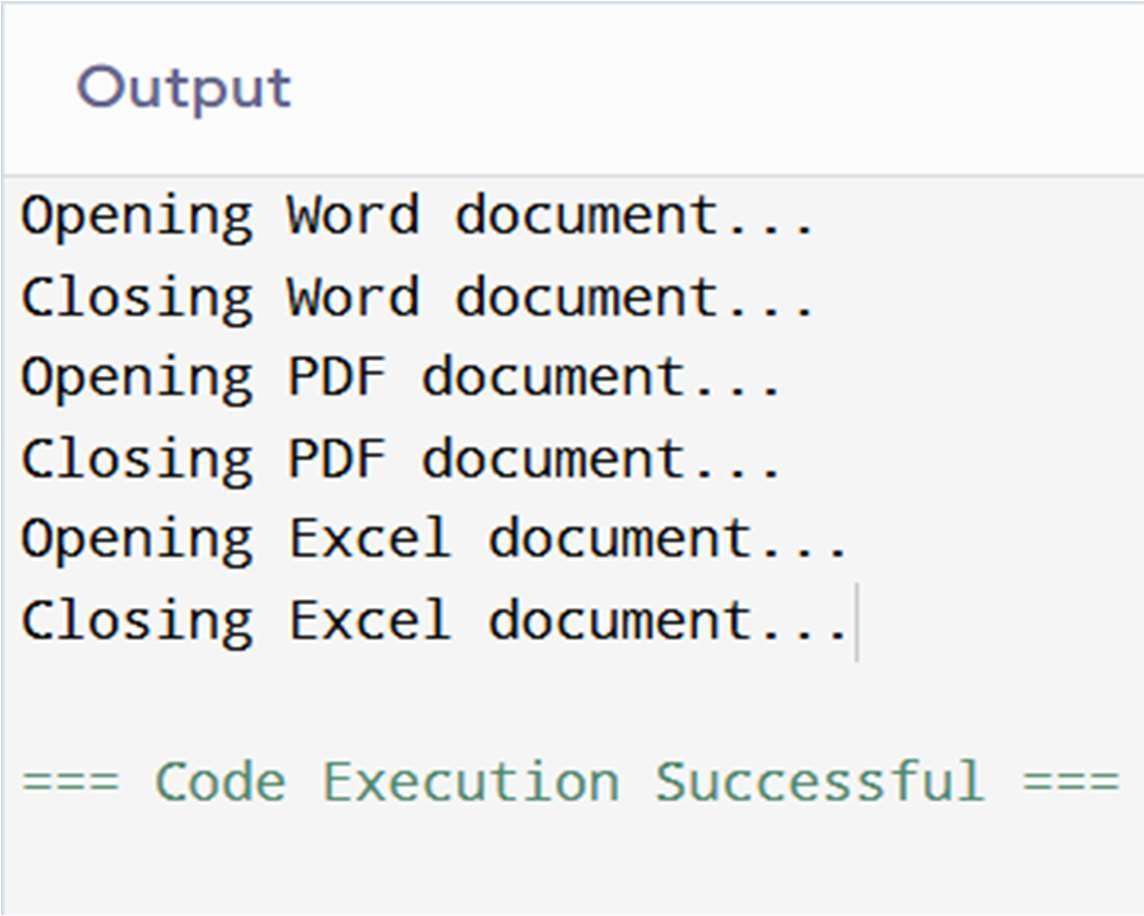
DocumentFactory excelFactory = new ExcelFactory(); Document excelDoc = excelFactory.createDocument(); excelDoc.open();

excelDoc.close();

}

}

**Output:**

****