**Module 1 : DESIGN PATTERNS AND PRINCIPLES**

**Exercise 1: Implementing the Singleton Pattern**

**Code:**

**Logger.java — Singleton Implementation :**

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 logger1 and logger2 are the same instance.”);

} else {

System.out.println(“Different instances exist. Singleton failed.”);

}

}

}

class Logger {

private static Logger instance = new Logger();

private Logger() {

System.out.println(“Logger initialized.”);

}

public static Logger getInstance() {

return instance;

}

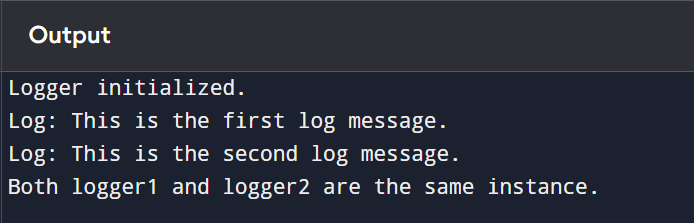
public void log(String message) {

System.out.println(“Log: “ + message);

}

}

**Output:**

****

**Exercise 2: Implementing the Factory Method Pattern**

**Code:**

**FactoryMethodPattern.java:**

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

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 {

public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

class ExcelDocumentFactory extends DocumentFactory {

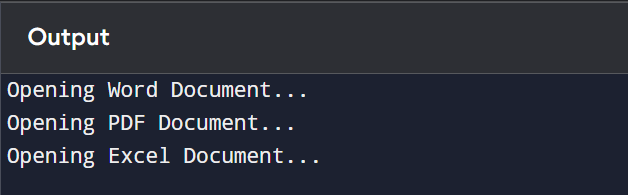
public Document createDocument() {

return new ExcelDocument();

}

}

**Output:**

****