**1)Singleton Pattern:**

Logger.java

public class Logger {

private static Logger instance;

private Logger() {

if (instance != null) {

throw new RuntimeException("Use getInstance");

}

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("test 1");

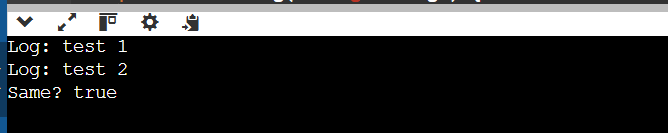
logger2.log("test 2");

System.out.println("Same? " + (logger1 == logger2));

}

}

**Output:**

****

**2)Factory Pattern:**

DocumentFactoryDemo.java

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

System.out.println("Word doc open");

}

}

class PdfDocument implements Document {

public void open() {

System.out.println("PDF doc open");

}

}

class ExcelDocument implements Document {

public void open() {

System.out.println("Excel doc open");

}

}

abstract class DocumentFactory {

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();

}

}

public class DocumentFactoryDemo {

public static void main(String[] args) {

DocumentFactory wordFac = new WordDocumentFactory();

DocumentFactory pdfFac = new PdfDocumentFactory();

DocumentFactory excelFac = new ExcelDocumentFactory();

Document word = wordFac.createDocument();

Document pdf = pdfFac.createDocument();

Document excel = excelFac.createDocument();

word.open();

pdf.open();

excel.open();

}

}

**Output:**

****