**Exercise 1: Implementing the Singleton Pattern**

CODE:

// Online Java Compiler

// Use this editor to write, compile and run your Java code online

class Main {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

if (logger1 == logger2) {

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

} else {

System.out.println("Different logger instances exist.");

}}}

class Logger {

private static Logger instance;

private Logger() {

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

}

public static Logger getInstance() {

if (instance == null) {

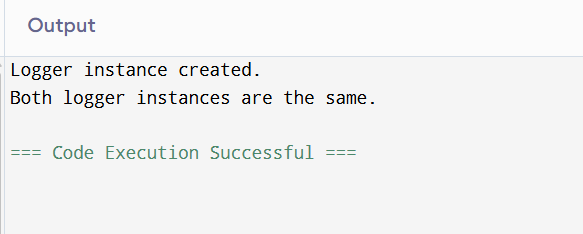
instance = new Logger();

}

return instance;

}}

OUTPUT:



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

CODE:

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

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

}}

class PdfDocument implements Document {

public void open() {

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

}}

class ExcelDocument implements Document {

public void open() {

System.out.println("Opening an 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();

}}

class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document word = wordFactory.createDocument();

word.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.createDocument();

pdf.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.createDocument();

excel.open();

}

}

OUTPUT:

