**Module 1: Design Patterns and Principles**

**Exercise 1: Singleton Pattern**

**Code:**

public class Logger {

private static Logger instance1;

private Logger()

{

System.out.println("A logger instance was created");

}

public static Logger get()

{

if(instance1==null)

{

instance1=new Logger();

}

return instance1;

}

public void log(String message) {

System.out.println("The message is: " + message);

}

public static void main(String[] args)

{

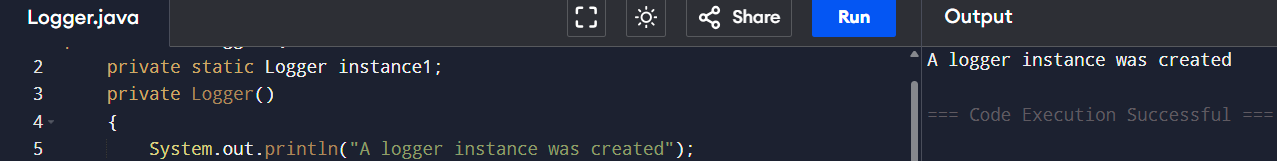
Logger a=Logger.get();

Logger b=Logger.get();

}

}

**Output:**



**Exercise 2: Factory Method pattern**

**Code:**

public class Main {

public interface Document {

void read();

}

public static class WordDocument implements Document {

public void read() {

System.out.println("I am performing read operation in Word document.");

}

}

public static class PdfDocument implements Document {

public void read() {

System.out.println("I am performing read operation in PDF document.");

}

}

public static class ExcelDocument implements Document {

public void read() {

System.out.println("I am performing read operation in Excel document.");

}

}

public static abstract class DocumentFactory {

public abstract Document createDocument();

}

public static class WordDocFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

public static class PdfDocFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

public static class ExcelDocFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.read();

DocumentFactory pdfFactory = new PdfDocFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.read();

DocumentFactory excelFactory = new ExcelDocFactory();

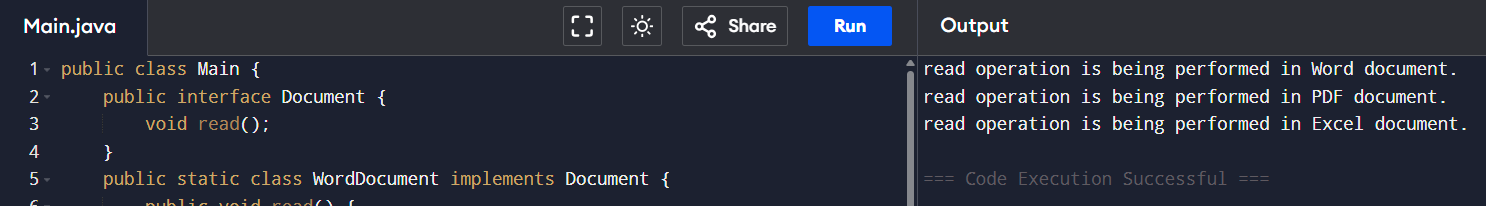
Document excelDoc = excelFactory.createDocument();

excelDoc.read();

}

}

**Output:**

****