Exercise 1: Implementing the Singleton Pattern

Logger.java:

class Logger {

      private static Logger instance;

      private Logger() {

      }

      public static Logger getInstance() {

            if (instance == null) {

                  instance = new Logger();

            }

            return instance;

      }

}

Test.java:

public class Test {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        if (logger1 == logger2) {

            System.out.println("Both logger1 and logger2 are the same instance.");

        } else {

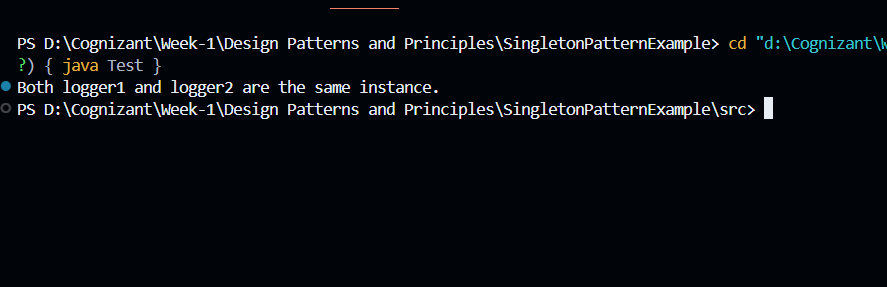
            System.out.println("logger1 and logger2 are different instances.");

        }

    }

}

Output:



Exercise 2: Implementing the Factory Method Pattern

Document.java:

public interface Document {

void open();

}

DocumentFactory.java

public abstract class DocumentFactory {

      public abstract Document createDocument();

}

ExcelDocument.java:

public class ExcelDocument implements Document {

      @Override

      public void open() {

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

      }

}

ExcelDocumentFactory.java:

public class ExcelDocumentFactory extends DocumentFactory {

      @Override

      public Document createDocument() {

            return new ExcelDocument();

      }

}

Main.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();

    }

}

PdfDocument.java:

public class PdfDocument implements Document {

      @Override

      public void open() {

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

      }

}

PdfDocumentFactory.java:

public class PdfDocumentFactory extends DocumentFactory {

      @Override

      public Document createDocument() {

            return new PdfDocument();

      }

}

WordDocument.java:

public class WordDocument implements Document {

      @Override

      public void open() {

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

      }

}

WordDocumentFactory.java:

public class WordDocumentFactory extends DocumentFactory {

      @Override

      public Document createDocument() {

            return new WordDocument();

      }

}

Output:

