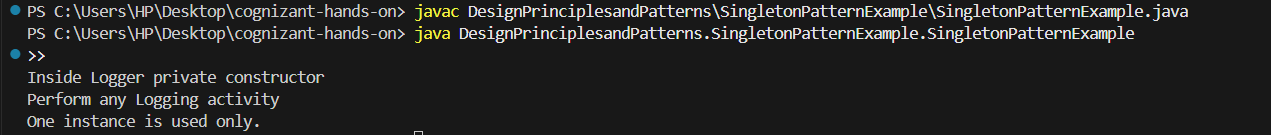
**Exercise 1: Implementing the Singleton Pattern**

|  |
| --- |
| package DesignPrinciplesandPatterns.SingletonPatternExample;  class Logger {      private static Logger obj;      private Logger() {          System.out.println("Inside Logger private constructor");      }      public static Logger getInstance() {          if (obj == null) {              obj = new Logger();          }          return obj;      }      public void LoggingActivityMethod() {          System.out.println("Perform any Logging activity");      }  }  class SingletonPatternExample {      public static void main(String[] args) {          Logger l1 = Logger.getInstance();          Logger l2 = Logger.getInstance();          l1.LoggingActivityMethod();          if (l1 == l2) {              System.out.println("One instance is used only.");          } else {              System.out.println("Different instances used.");          }      }  } |

**OUTPUT:**

****

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

|  |
| --- |
| package DesignPrinciplesandPatterns.SingletonPatternExample.FactoryMethodPatternExample;  interface Document {      void openDoc();  }  class WordDocument implements Document {      public void openDoc() {          System.out.println("Try opening a word file.");      }  }  class PdfDocument implements Document {      public void openDoc() {          System.out.println("Try opening a pdf file.");      }  }  class ExcelDocument implements Document {      public void openDoc() {          System.out.println("Try opening an excel file.");      }  }  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();      }  }  public class FactoryMethodPatternExample {      public static void main(String[] args) {          DocumentFactory wordFactory = new WordDocumentFactory();          Document wordDoc = wordFactory.createDocument();          wordDoc.openDoc();          DocumentFactory pdfFactory = new PdfDocumentFactory();          Document pdfDoc = pdfFactory.createDocument();          pdfDoc.openDoc();          DocumentFactory excelFactory = new ExcelDocumentFactory();          Document excelDoc = excelFactory.createDocument();          excelDoc.openDoc();      }  } |

**OUTPUT- :**

