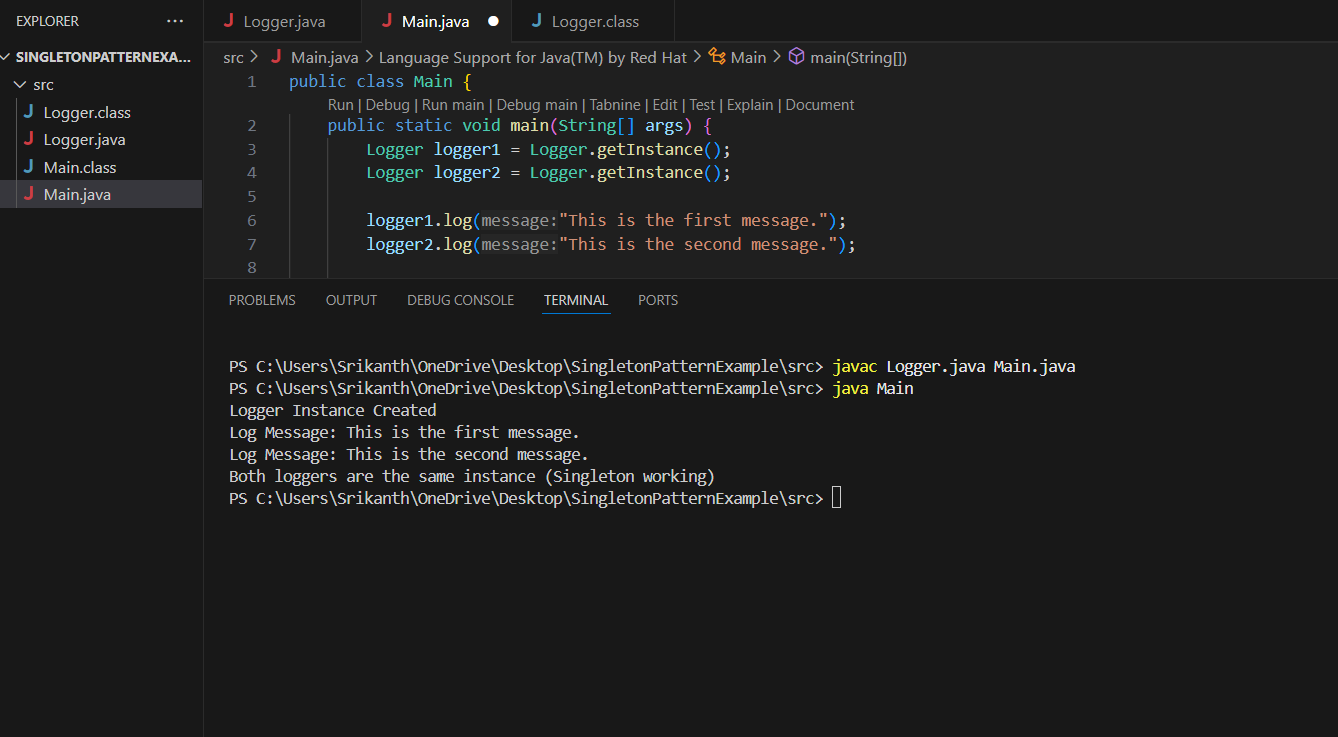
**Module 1(Design Patterns and Principles)**

**Exercise 1: Implementing the Singleton Pattern**

**Solution Code:**

|  |  |
| --- | --- |
| Logger.java  public 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;      }      public void log(String message) {          System.out.println("Log Message: " + message);      }  } | Main.java  public class Main {      public static void main(String[] args) {          Logger logger1 = Logger.getInstance();          Logger logger2 = Logger.getInstance();          logger1.log("This is the first message.");          logger2.log("This is the second message.");          if (logger1 == logger2) {              System.out.println("Both loggers are the same instance (Singleton working)");          } else {              System.out.println("Different instances");          }      }  } |

**Output:**



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

**Solution Code:**

|  |  |
| --- | --- |
| Document.java  public interface Document {      void open();  } | WordDocument.java  public class WordDocument implements Document {      public void open() {          System.out.println("Opening a Word document.");      }  } |
| PdfDocument.java  public class PdfDocument implements Document {      public void open() {          System.out.println("Opening a PDF document.");      }    } | ExcelDocument.java  public class ExcelDocument implements Document{      public void open() {          System.out.println("Opening an Excel document.");      }  } |
| DocumentFactory.java  public abstract class DocumentFactory {      public abstract  Document createDocument();  } | WordDocumentFactory.java  public class WordDocumentFactory extends DocumentFactory {       public Document createDocument() {          return new WordDocument();      }  } |
| PdfDocumentFactory.java  public class PdfDocumentFactory extends DocumentFactory {      public Document createDocument() {          return new PdfDocument();      }  } | ExcelDocumentFactory.java  public class ExcelDocumentFactory extends DocumentFactory{      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();      }    } |

**Output:**

