**Exercise 1: Implementing the Singleton Pattern**

**Code:**

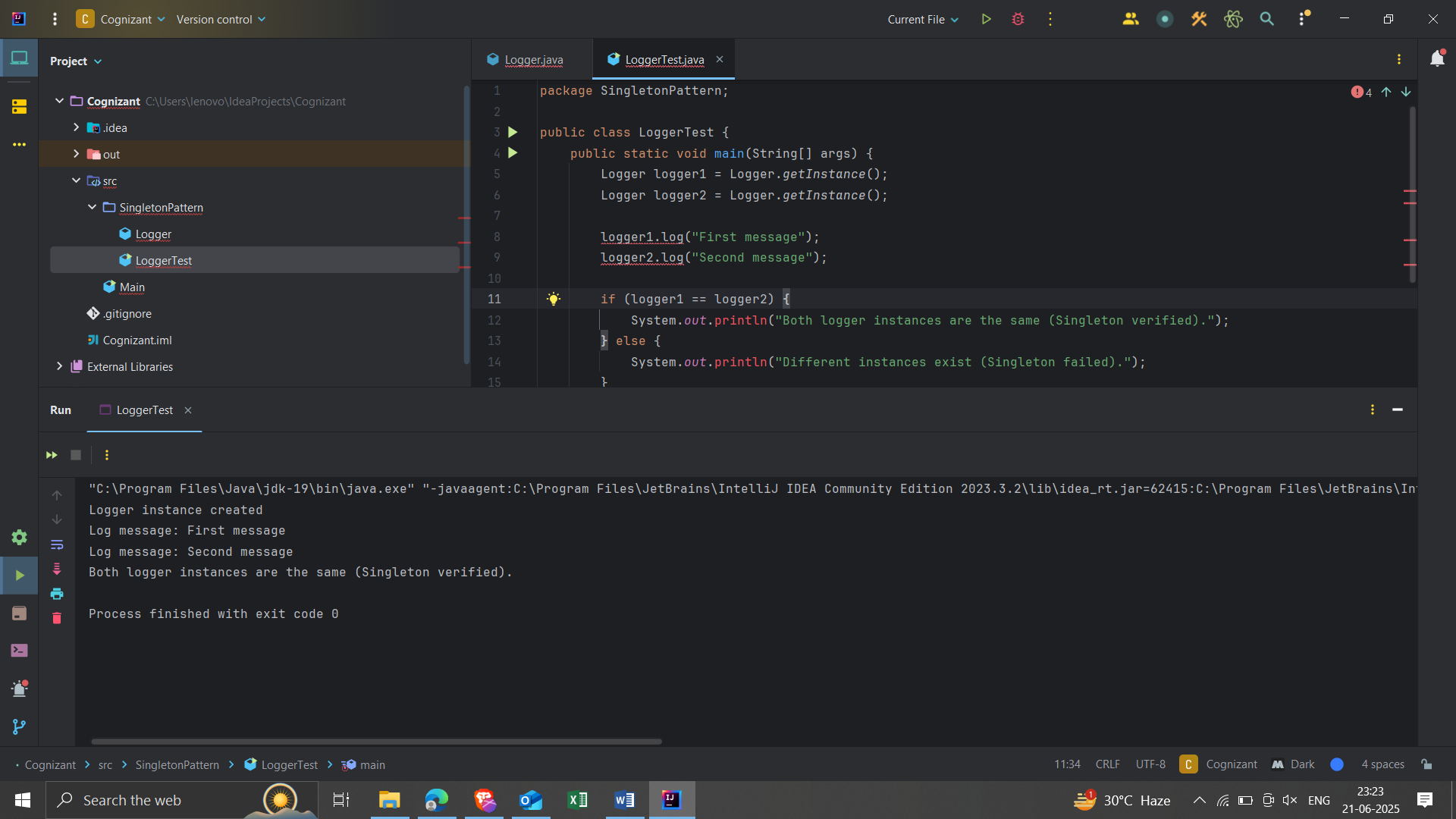
Logger.java

package SingletonPattern;  
  
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);  
 }  
}

LoggerTest.java

package SingletonPattern;  
  
public class LoggerTest {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 Logger logger2 = Logger.*getInstance*();  
  
 logger1.log("First message");  
 logger2.log("Second message");  
  
 if (logger1 == logger2) {  
 System.*out*.println("Both logger instances are the same (Singleton verified).");  
 } else {  
 System.*out*.println("Different instances exist (Singleton failed).");  
 }  
 }  
}

**Output:**



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

**Code:**

Document.java

package factory;  
  
public interface Document {  
 void open();  
}

WordDocument.java

package factory;  
  
public class WordDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening a Word document.");  
 }  
}

PdfDocument.java

package factory;  
  
public class PdfDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening a PDF document.");  
 }  
}

ExcelDocument.java

package factory;  
  
public class ExcelDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening an Excel document.");  
 }  
}

DocumentFactory.java

package factory;  
  
public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}

WordDocumentFactory.java

package factory;  
  
public class WordDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}

PdfDocumentFactory.java

package factory;  
  
public class PdfDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new PdfDocument();  
 }  
}

ExcelDocumentFactory.java

package factory;  
  
public class ExcelDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new ExcelDocument();  
 }  
}

FactoryPatternTest.java

package factory;  
  
public class FactoryPatternTest {  
 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:**

