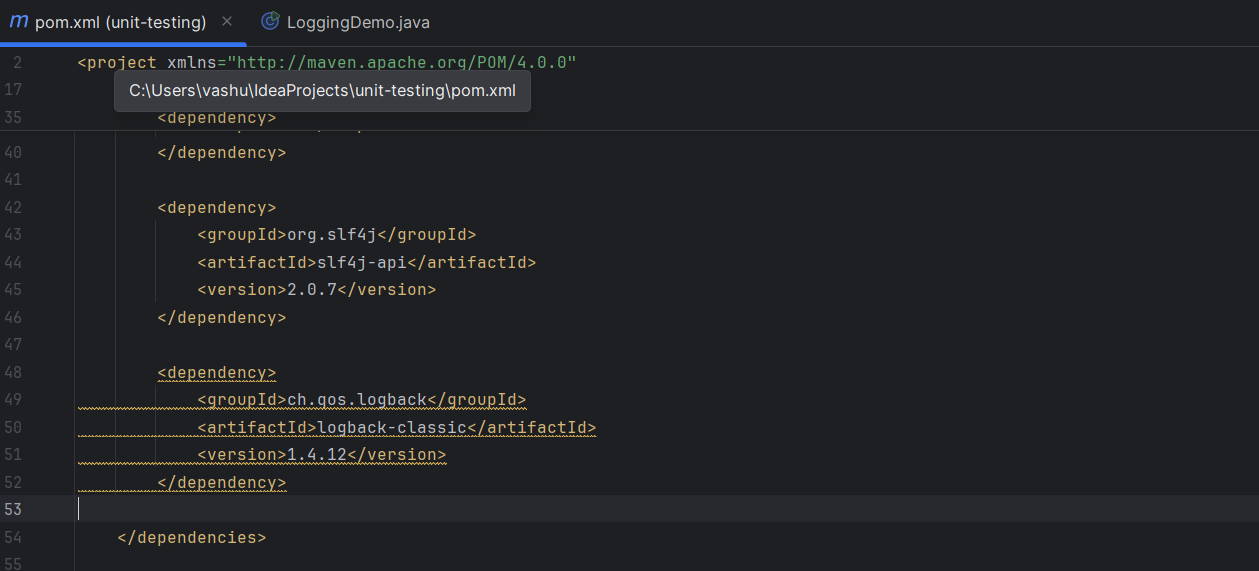
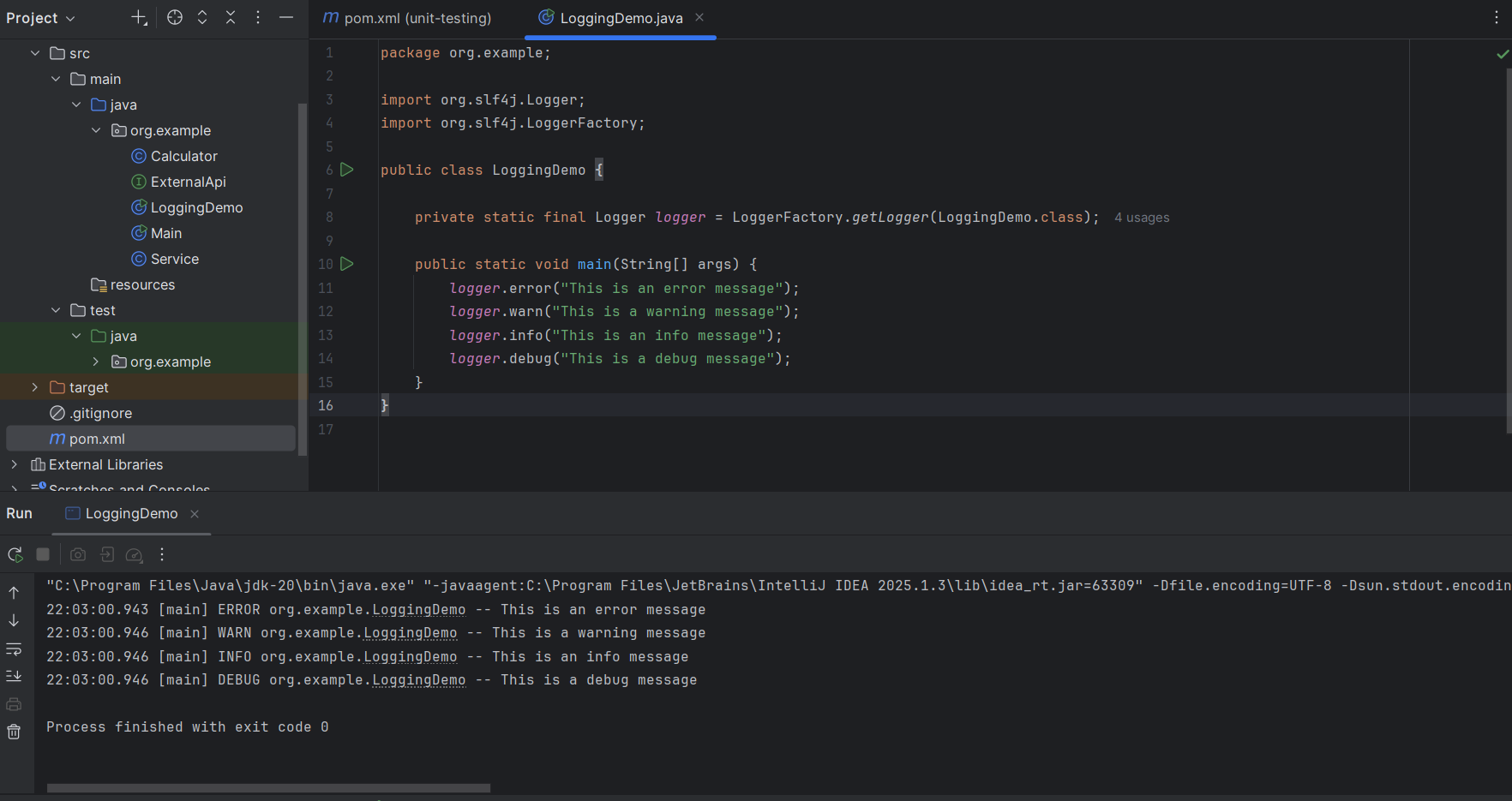
**Exercise 1: Logging Error Messages and Warning Levels**

* **Working Steps**
* Add SLF4J and Logback Dependencies - In Maven, you need to add these to your pom.xml file.
* Create a Java Class with SLF4J Logging
* After setting up the SLF4J and Logback configuration and writing your LoggingExample class, you need to run the Java program to see the logging output.

****

****

* **Terminologies/Syntax Used**

**🔷 SLF4J**

SLF4J stands for *Simple Logging Facade for Java*. It acts as a common interface for various logging frameworks such as Logback, Log4j, and java.util.logging. This allows developers to write logging code that is independent of the actual logging implementation.

**🔷 Logback**

Logback is a modern and flexible logging framework that is fully compatible with SLF4J. It is often used as the default implementation when working with SLF4J and is considered the successor to Log4j.

**🔷 logger.error()**

The error() method is used to log error messages that indicate a critical problem in the application. These messages typically represent issues that might cause the application to crash or behave incorrectly.

**🔷 logger.warn()**

The warn() method is used to log warning messages. These messages signal potential issues or situations that could lead to problems in the future, but they don’t immediately disrupt the program’s flow.

**🔷 logger.info()**

The info() method is used to log informational messages that highlight the normal operation and progress of the application. These messages are useful for understanding the general flow of execution during normal usage.

**🔷 logger.debug()**

The debug() method is used to log detailed messages primarily intended for debugging purposes during development. These messages usually include low-level information about variables, control flow, and logic, and are typically disabled in production environments to reduce log clutter.