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

**Task:**  
Write a Java application that demonstrates logging error messages and warning levels using SLF4J.

**Step 1: Add to pom.xml:**

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

**Step 2: Java Code**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

public static void main(String[] args) {

logger.error("This is an error message");

logger.warn("This is a warning message");

}

}

**Exercise 2: Parameterized Logging**

**Task:**  
Write a Java application that demonstrates parameterized logging using SLF4J.

**Step 1: Add to pom.xml:**  
(Same as above)

**Step 2: Java Code**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class ParameterizedLogging {

private static final Logger logger = LoggerFactory.getLogger(ParameterizedLogging.class);

public static void main(String[] args) {

String username = "Janani";

int userId = 101;

logger.info("User {} with ID {} has logged in", username, userId);

logger.warn("Low balance for user: {}", username);

logger.error("User {} encountered an error while processing payment", username);

}

}

**Exercise 3: Using Different Appenders**

**Task:**  
Write a Java application that demonstrates using different appenders with SLF4J.

**Step 1: Add to pom.xml:**  
(Same as above)

**Step 2: Create logback.xml in src/main/resources:**

<configuration>

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">

<encoder>

<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>

</encoder>

</appender>

<appender name="file" class="ch.qos.logback.core.FileAppender">

<file>logs/app.log</file>

<append>true</append>

<encoder>

<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>

</encoder>

</appender>

<root level="debug">

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>

Make sure logs/ folder exists or will be created in the project directory.

**Step 3: Java Code**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppenderExample {

private static final Logger logger = LoggerFactory.getLogger(AppenderExample.class);

public static void main(String[] args) {

logger.info("This is an informational message logged to both console and file.");

logger.debug("Debug message for developer visibility.");

logger.error("Error message for system failure reporting.");

}

}