**WEEK-2**

**SL4J Logging Exercises**

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

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

Step-by-Step Solution:

1. Add SLF4J and Logback dependencies to your 'pom.xml file:

<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>

2. Create a Java class that uses SI.F4] for logging:

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");

}

}

**Code:**

**Steps:**

1.Create Maven Project with group id: com.example and artifact id: SLF4JDemo

2. Add SLF4J + Logback Dependencies to pom.xml

<dependencies>

<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>

</dependencies>

**3. LoggingEx.java**

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingEx{

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

public static void main(String[] args) {

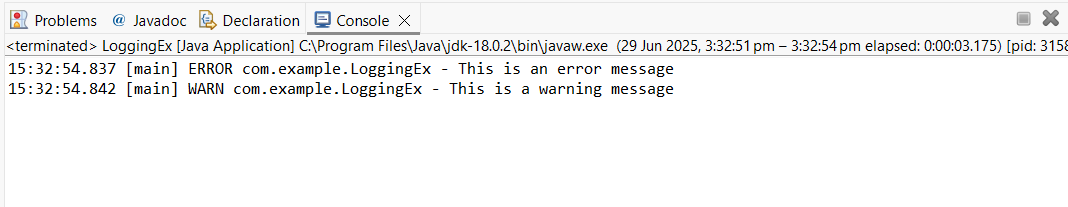
logger.error("This is an error message");

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

}

}

**Output:**



**Explanation:**

* We used SLF4J, a logging interface that is compatible with several logging frameworks, in conjunction with Logback.
* I updated the pom.xml file to include two dependencies: one for logback-classic and one for slf4j-api.
* I used LoggerFactory to construct a logger in my code. main() made use of getLogger().
* Clear error and warning messages are printed in the console by the logger.
* This configuration facilitates the effective tracking of errors, problems, or warnings during runtime in real-world projects.