**WEEK – 2 SLF4J LOGGING FRAMEWORK**

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

A Java application that demonstrates logging error messages and warning levels using SLF4J.

**DEPENDENCY:**

<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>

These dependencies do two things:

* The **SLF4J API** allows you to write generic logging code without binding it to a specific implementation.
* The **Logback Classic** dependency is the actual logging engine that takes care of formatting and writing the log messages.

**APPLOGGER.JAVA**

package com.assignment;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppLogger {

private static final Logger log = LoggerFactory.getLogger(AppLogger.class);

public static void main(String[] args) {

log.error("Critical failure: Unable to connect to the database.");

log.warn("Configuration warning: Using default settings.");

}

}

* It uses SLF4J to get a logger instance tied to the current class (AppLogger).
* It logs two types of messages:  
  • An **error message**, representing a serious system failure.  
  • A **warning message**, indicating a non-critical but abnormal configuration condition.

**Output:**

14:30:42.110 [main] ERROR com.assignment.AppLogger - Critical failure: Unable to connect to the database.

14:30:42.113 [main] WARN com.assignment.AppLogger - Configuration warning: Using default settings.