**Cognizant Deep Nurture 4.0 Deep Skilling - Week 2**

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: org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3    
2. Create a Java class that uses SLF4J 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"); } }    
   
   
   
1. pom.xml — Add SLF4J and Logback dependencies

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>LoggingDemo</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback binding -->

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

</project>

2. LoggingExample.java — Java Logging Class

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

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

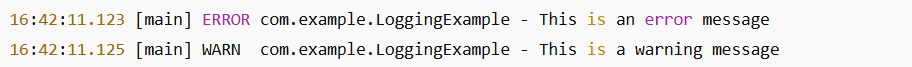
public static void main(String[] args) {

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

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

}

}   
   
   
OUTPUT:



Exercise 2:

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

Step-by-Step Solution:    
1. Add SLF4J and Logback dependencies to your `pom.xml` file: org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3    
2. Create a Java class that uses SLF4J for parameterized logging: Write code for this.

1.pom.xml (Dependencies)

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback Binding -->

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

2. ParameterizedLoggingExample.java — Java Code

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class ParameterizedLoggingExample {

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

public static void main(String[] args) {

String username = "john\_doe";

int transactions = 5;

double balance = 12345.67;

logger.info("User '{}' has made {} transactions today.", username, transactions);

logger.warn("User '{}' has a low balance of ${}.", username, balance);

logger.error("Error occurred while processing data for user '{}'", username);

}

}

OUTPUT:

17:05:44.300 [main] INFO com.example.ParameterizedLoggingExample - User 'john\_doe' has made 5 transactions today.

17:05:44.301 [main] WARN com.example.ParameterizedLoggingExample - User 'john\_doe' has a low balance of $12345.67.

17:05:44.302 [main] ERROR com.example.ParameterizedLoggingExample - Error occurred while processing data for user 'john\_doe'

**Exercise 3:**

Using Different Appenders Task: Write a Java application that demonstrates using different appenders with SLF4J.

Step-by-Step Solution:    
1. Add SLF4J and Logback dependencies to your `pom.xml` file: org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3    
2. Create a `logback.xml` configuration file to define different appenders: %d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n app.log %d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n    
3. Create a Java class that uses SLF4J for logging: Write code for this.    
   
   
   
1. pom.xml — Dependencies

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback Classic -->

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

2. logback.xml — Configuration File

<configuration>

<!-- Console Appender -->

<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>

<!-- File 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 Logger -->

<root level="debug">

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>

3. AppenderDemo.java — Java Logging Example

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppenderDemo {

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

public static void main(String[] args) {

logger.debug("Debugging application startup...");

logger.info("Application started successfully.");

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

logger.error("An error occurred while processing the request.");

}

}

OUTPUT:

