***WEEK 2***

***Logging using SLF4J***

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

**LoggingExample.java**

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");

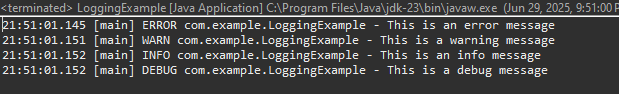
***logger***.info("This is an info message");

***logger***.debug("This is a debug message");

}

}

**OUTPUT**

****

**Exercise 2: Parameterized Logging**

**ParameterizedLoggingExample.java**

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 = "Indhumathi";

int items = 5;

double price = 123.45;

***logger***.info("User {} has purchased {} items worth ₹{}", username, items, price);

***logger***.debug("Debugging purchase: username={}, items={}, price={}", username, items, price);

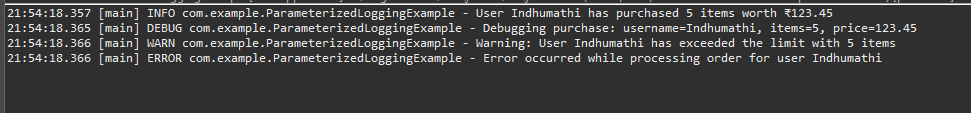
***logger***.warn("Warning: User {} has exceeded the limit with {} items", username, items);

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

}

}

**OUTPUT**

****

**Exercise 3: Using Different Appenders**

**LoggingWithAppenders.java**

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingWithAppenders {

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

public static void main(String[] args) {

***logger***.debug("Debugging the application...");

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

***logger***.warn("Low disk space warning");

***logger***.error("An unexpected error occurred");

}

}

**Logback.xml**

<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>app.log</file>

<append>true</append>

<encoder>

<pattern>%d{yyyy-MM-dd HH:mm:ss} %-5level %logger{36} - %msg%n</pattern>

</encoder>

</appender>

<!-- Root Logger uses both -->

<root level="debug">

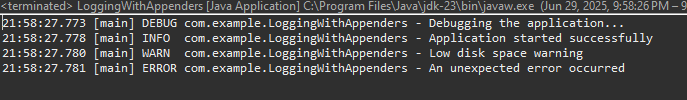
<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>

**OUTPUT**

****