**Logging using SLF4J**

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

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

using SLF4J.

package com.example.junit\_demo;

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) {

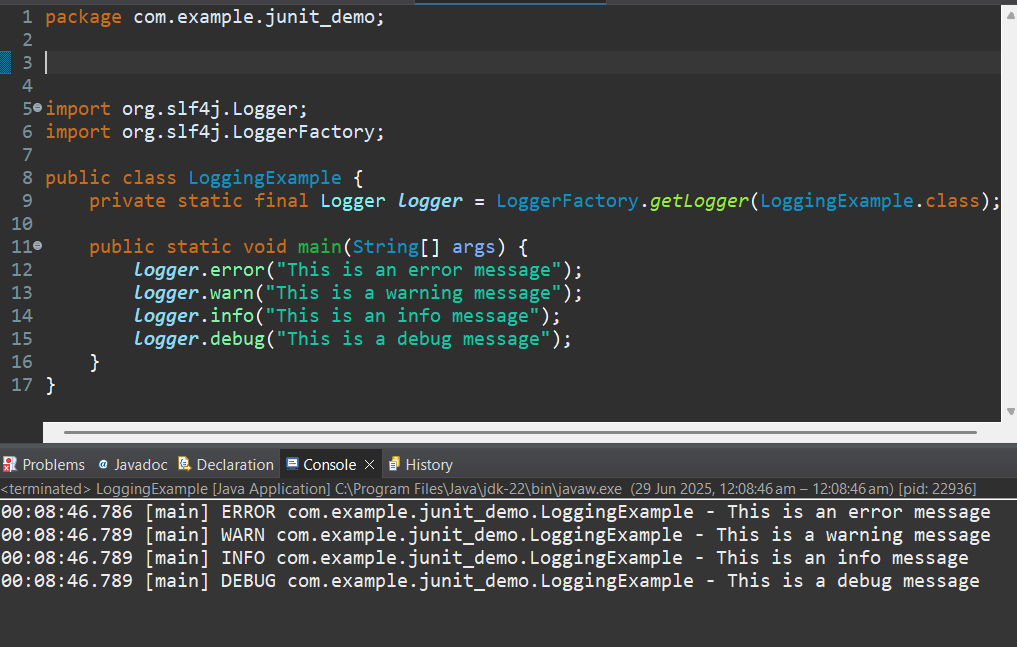
***logger***.error("This is an error message");

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

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

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

}

}

**Exercise 2: Parameterized Logging**

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

**ParameterizedLogging**

package com.example.junit\_demo;

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";

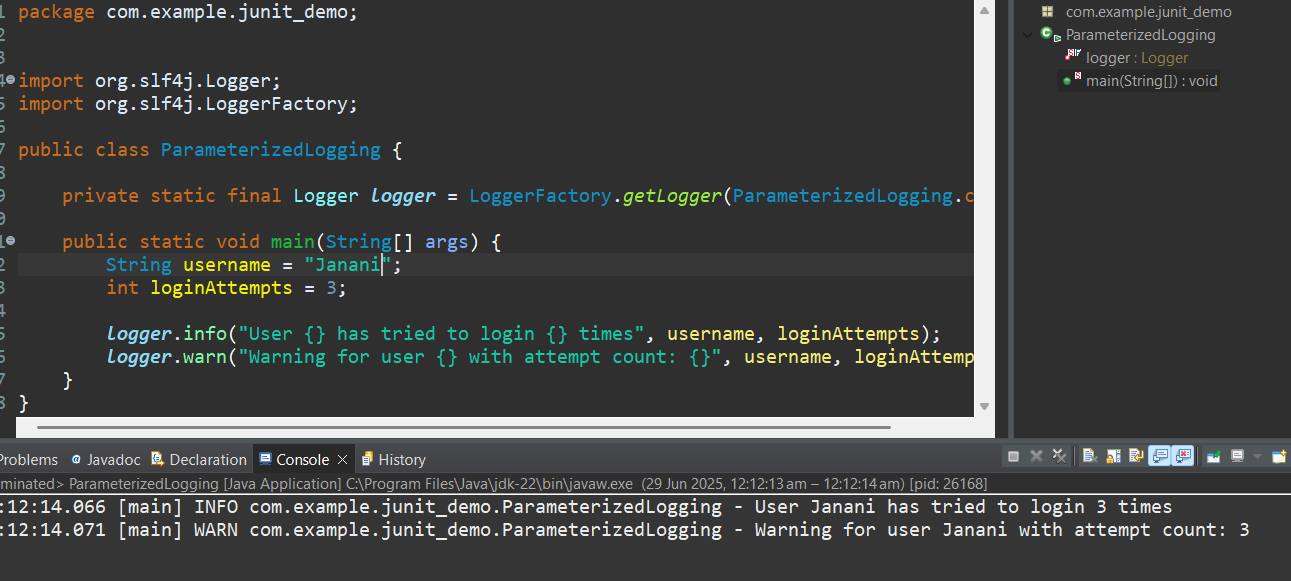
int loginAttempts = 3;

***logger***.info("User {} has tried to login {} times", username, loginAttempts);

***logger***.warn("Warning for user {} with attempt count: {}", username, loginAttempts);

}

}



**Exercise 3: Using Different Appenders**

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

LoggingExample.java

package com.example.junit\_demo;

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");

}

}

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>logs/app.log</file> <!-- or absolute path like C:/logs/app.log -->

<append>true</append>

<encoder>

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

</encoder>

</appender>

<!-- Root Logger -->

<root level="debug">

<appender-ref ref="console"/>

<appender-ref ref="file"/>

</root>

</configuration>

