**Exercise-1**

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");

logger.trace("This is a trace message");

}

}

**Exercise-2**

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 user = "John";

int age = 30;

// Parameterized logging examples

logger.info("User {} is {} years old", user, age);

logger.debug("Processing request for user: {}", user);

logger.error("Failed to process request for user: {}. Error code: {}", user, 404);

// Conditional logging

if (logger.isDebugEnabled()) {

logger.debug("Detailed debug information: {}", getDetailedDebugInfo());

}

}

private static String getDetailedDebugInfo() {

return "Some detailed debug information";

}

}

**Exercise-3**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppenderExample {

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

public static void main(String[] args) {

logger.trace("This is a trace message - should not appear with default config");

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

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

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

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

try {

int result = 10 / 0;

} catch (Exception e) {

logger.error("Exception occurred: ", e);

}

}

}

also need the logback.xml configuration file

<configuration>

<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>

<appender name="file" class="ch.qos.logback.core.FileAppender">

<file>app.log</file>

<encoder>

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

</encoder>

</appender>

<root level="debug">

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>