**Solution:**

**pom.xml**

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.logging</groupId>

<artifactId>SLF4J</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback (SLF4J Implementation) -->

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

</project>

**LoggingExample.java**

package com.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");

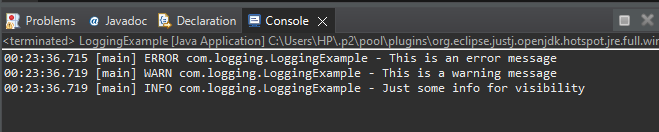
logger.warn("This is a warning message");

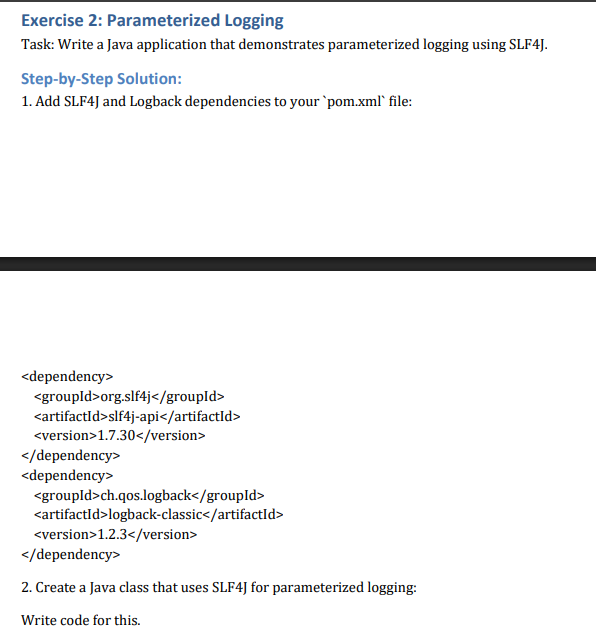
logger.info("Just some info for visibility");

}

}

**Output:**





**Solution:**

**ParameterizedLogging.java**

package com.logging;

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 = "Satyam";

int loginCount = 5;

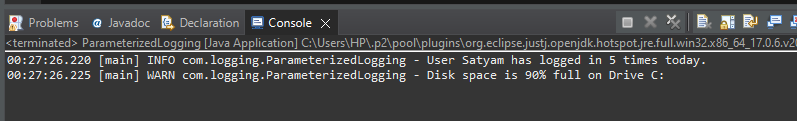
logger.info("User {} has logged in {} times today.", username, loginCount);

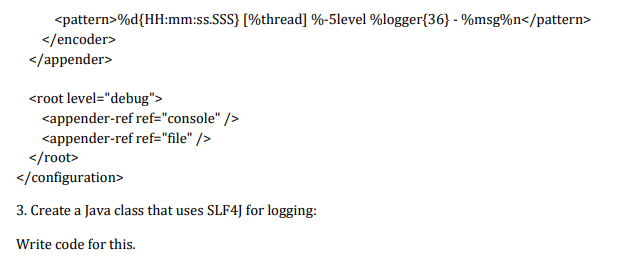
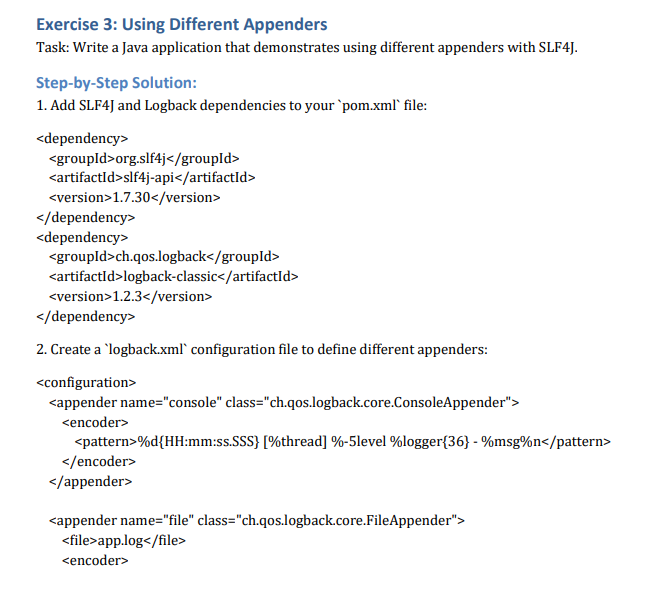
logger.warn("Disk space is {}% full on {}", 90, "Drive C:");

}

}

**Output:**





**Solution:**

**AppenderExample.java**

package com.logging;

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.info("This goes to console and file both.");

logger.debug("Debug level log will appear too.");

logger.error("Simulating error for both outputs.");

}

}

**Output:**

