**“Logging using SLF4J”**

**Exercise 1: Logging Error Messages and Warning Level , Write a Java application that demonstrates logging error messages and warning levels using SLF4J.**

***SOLUTION:***

**1.LoggingExample.java**

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");

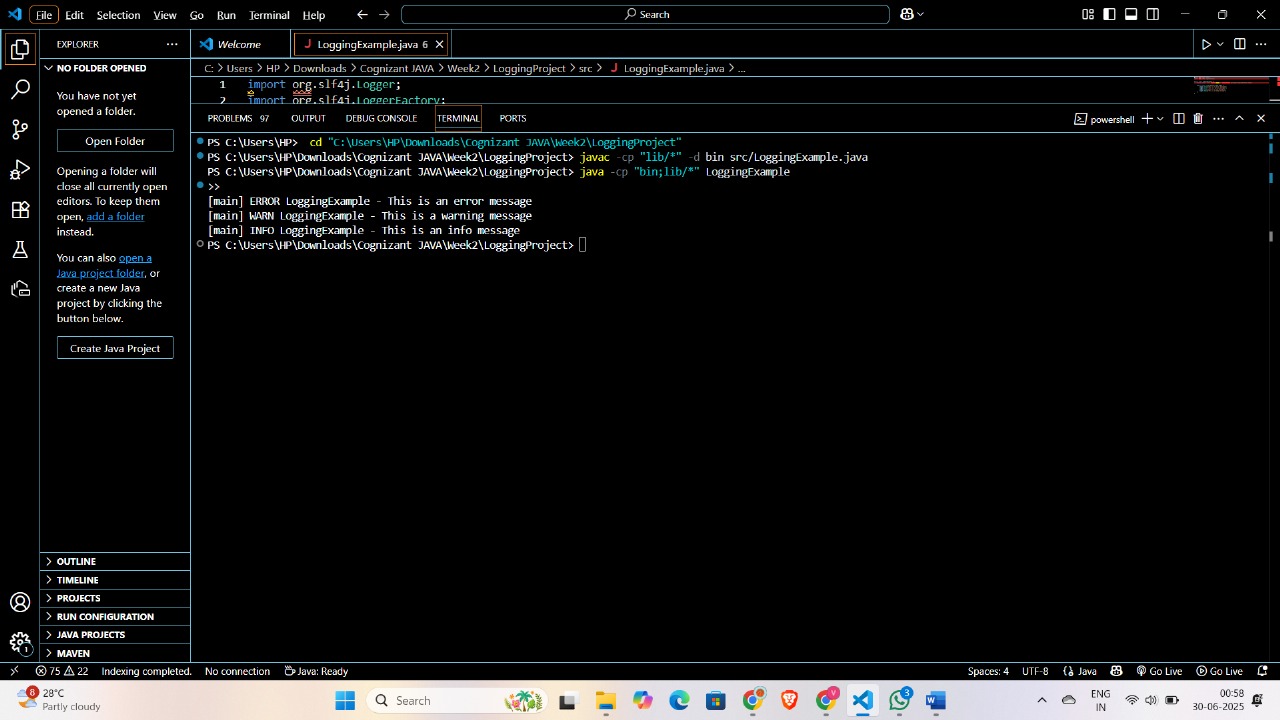
*logger*.warn("This is a warning message");

*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.**

***SOLUTION:***

**1.ParametrizedLoggingExample.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.time.LocalDateTime;

public class ParametrizedLoggingExample {

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

public static void main(String[] args) {

String username = "Gunashree";

String operation = "Login";

LocalDateTime timestamp = LocalDateTime.*now*();

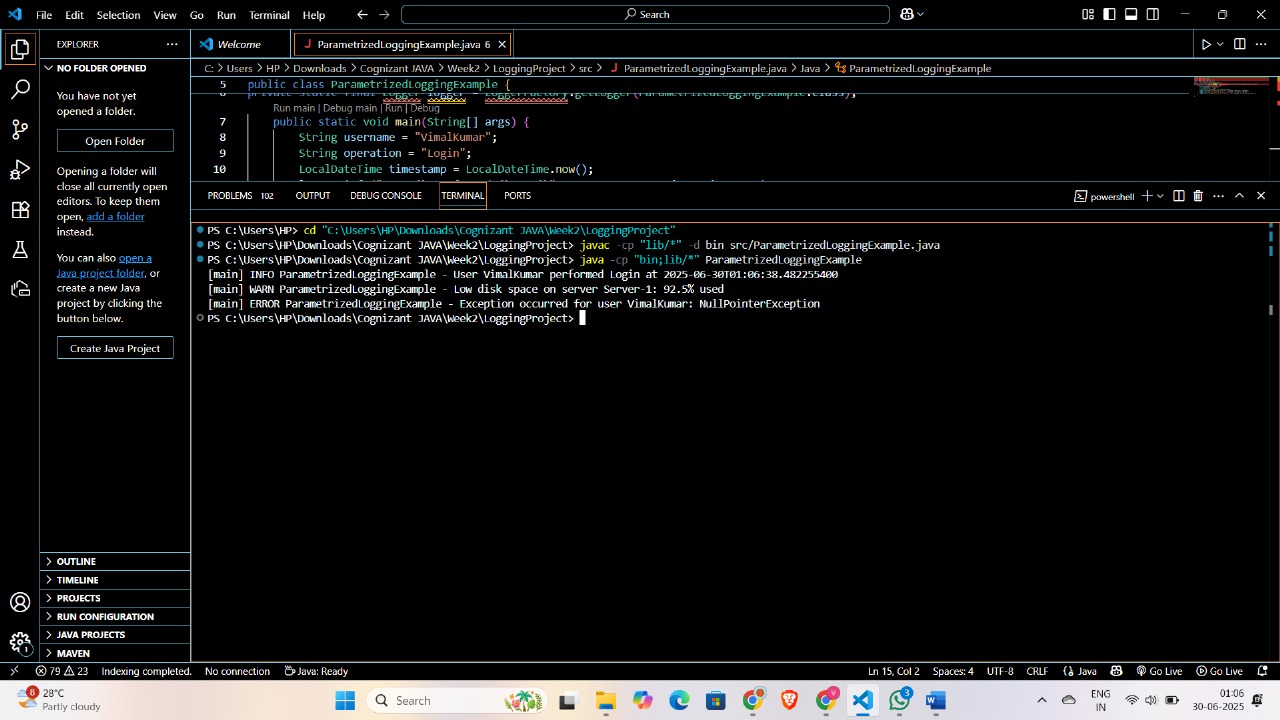
*logger*.info("User {} performed {} at {}", username, operation, timestamp);

*logger*.warn("Low disk space on server {}: {}% used", "Server-1", 92.5);

*logger*.error("Exception occurred for user {}: {}", username, "NullPointerException");

}

}



**---------------------------------------------------------------------------------------------------------------------------**

**Exercise 3: Using Different Appenders Task: Write a Java application that demonstrates using different appenders with SLF4J.**

***SOLUTION:***

**1.LoggerExample.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggerExample {

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

public static void main(String[] args) {

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

*logger*.debug("Debug level message");

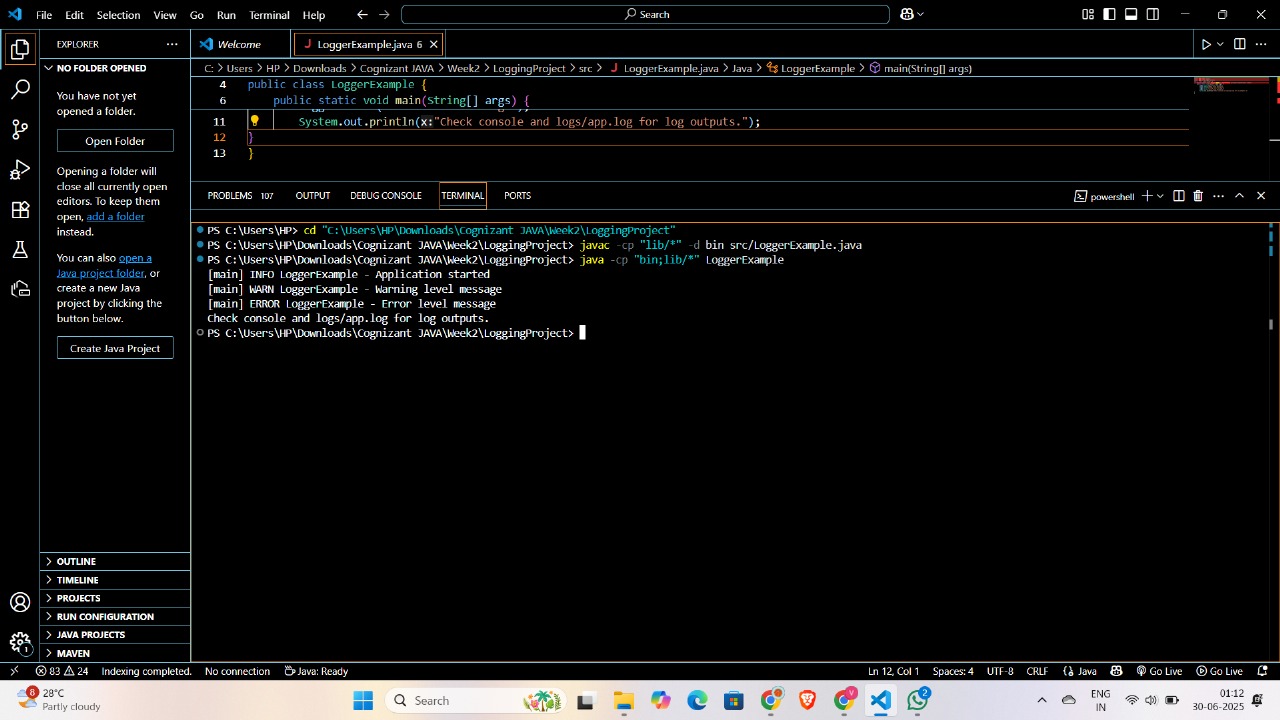
*logger*.warn("Warning level message");

*logger*.error("Error level message");

System.*out*.println("Check console and logs/app.log for log outputs.");

}

}



**---------------------------------------------------------------------------------------------------------------------------**