**Student Name:** KARTHYAYENI P

**Registration No:** 22CSR089

**Course/Batch:** KONGU ENGINEERING COLLEGE (B.E COMPUTER SCIENCE AND ENGINEERING)

**EXERCISE 1: LOGGING ERROR MESSAGES AND WARNING LEVELS**

**Introduction:**

This Java program demonstrates how to log error and warning messages using SLF4J (Simple Logging Facade for Java) in a temperature monitoring system. It simulates sensor data and provides meaningful logs for system diagnostics.

**Objective:**

* **Log Critical Failures:** Capture and log errors such as sensor failure using logger.error() to alert developers or monitoring tools.
* **Warn About Risk Conditions:** Use logger.warn() to notify about high temperature, low battery, or overused sensors that require attention.
* **Improve Observability:** Provide real-time feedback through logs to monitor system health and preempt potential issues.

**Implementation Breakdown:**

**TemperatureMonitor.java:**

package org.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class TemperatureMonitor {

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

public static void main(String[] args) {

double temperature = readSensor();

boolean batteryLow = true;

boolean sensorOverused = true;

if (temperature == -1) {

logger.error("Sensor failure detected. Unable to read temperature.");

System.out.println("Error: Could not read temperature.");

} else {

if (temperature > 40) {

logger.warn("High temperature detected: {}°C", temperature);

}

if (batteryLow) {

logger.warn("Sensor battery is low. Please replace it soon.");

}

if (sensorOverused) {

logger.warn("Sensor has exceeded recommended usage hours.");

}

System.out.println("Current temperature: " + temperature + "°C");

}

}

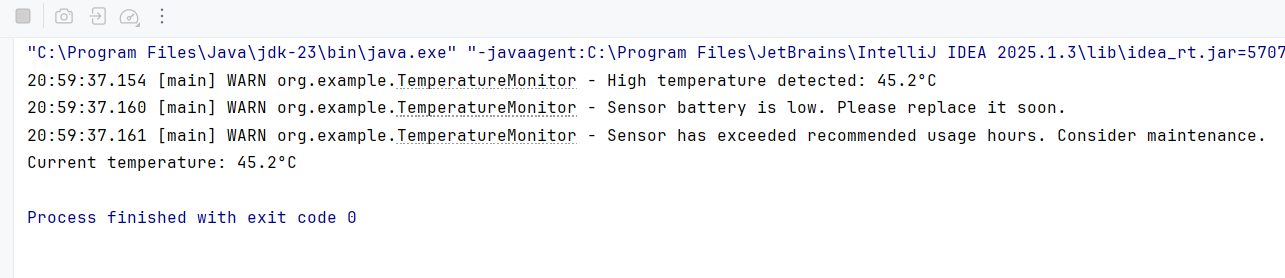
private static double readSensor() {

return 45.2;

}

}

**Output:**

****

**Conclusion:**

The program effectively demonstrates how SLF4J can be used to differentiate between error and warning levels in a real-world scenario. This structured logging aids in quick debugging, system monitoring, and maintaining reliable software performance.