**SL4J LOGGING EXERCISES**

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

This Java program shows how to log issue and warning messages utilizing SLF4J (Simple Logging Facade for Java) in a weather monitoring system. It mimics weather sensor readings and produces informative logs to assist in system diagnostics and environmental alerting.

**Goal:**

* Log Critical Failures  
  Capture and log issues such as API outages or sensor read malfunctions utilizing logger.issue() to alert system administrators or monitoring dashboards.
* Warn About Risk Conditions  
  Use logger.warn() to notify about high temperatures, low humidity, or freezing conditions that may require utilizer intervention.
* Improve Observability  
  Provide real-time visibility into system conditions through structured logs, enabling better monitoring, debugging, and proactive system maintenance.

**Code & Output:**

WeatherMonitor.java

package org.example;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class WeatherMonitor {  
  
 private static final Logger logger = LoggerFactory.getLogger(WeatherMonitor.class);  
  
 public static void main(String[] args) {  
 double temperature = readTemperatureSensor();  
 double humidity = readHumiditySensor();  
 boolean apiDown =true;   
  
 System.out.println("Weather Report:");  
 System.out.println("Temperature: " + temperature + "°C");  
 System.out.println("Humidity: " + humidity + "%");  
  
 if (apiDown) {  
 logger.error("Weather API is not responding...");  
 System.out.println("Error: Cannot retrieve live weather data.");  
 }  
  
 if (temperature > 40) {  
 logger.warn("High temperature detected: {}°C", temperature);  
 }  
  
 if (humidity < 20) {  
 logger.warn("Low humidity level detected: {}%", humidity);  
 }  
 }  
  
 private static double readTemperatureSensor() {  
 return 42.8; // Simulated value  
 }  
  
 private static double readHumiditySensor() {  
 return 15.5; // Simulated value  
 }  
}

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