

TCSS 360 Software Development and Quality Assurance Techniques (Summer 2020 - Section A)

Project Assignment #01 (PRJ#01-W03Mon)

Revision History

Revision 0 – posted on Mon June 22nd, 2020.

Submit PRJ#01A by 11:59 pm on Mon June 29th, 2020

Submit PRJ#01B by 11:59 pm on Mon July 6th, 2020

You will have two (2) days of grace period for this assignment.

General Information

- You can go with the randomly assigned group leader (or moderator) or choose somebody else who will volunteer for assuming the responsibilities of this position.
 - The group leader will be responsible for coordinating the group meetings, task assignments, and collecting and submitting the project deliverables at the end.
 - The group leader will be awarded +5 bonus points for her/his services for completing this assignment.
- You are going to develop a single software as a group in the specified period of the assignment.
 - You can make up to three (3) submissions for each assignment before it is closed, and only the last submission will be graded.
 - If you want to resubmit, please do so. You do not need to notify me.

PRJ#01A (W02-Mon) Checkpoint

- This is an interim checkpoint where you will individually answer questions about the progress of your project in the previous week. Your answers will help me to monitor your progress and to share some advice with your group in some cases to help your group progress.
- You will answer personal questions such as: What did you plan to do last week? How did you do last week? What went wrong/right the previous week? What are you planning to do next week? How was the performance of your group last week? What are the risks that may cause the group to miss the deadline?

PRJ#01B (W03-Mon) Final Delivery

- You will individually answer questions about your project experience and report the lessons you learned during this group assignment. You will also do a peer review of your group members.
- You will submit your group work. Only one person from your group will submit your group project on behalf of all group members.
 - [PRJ#01-GroupNN-src.zip](#): Your zipped project folder that includes all your source codes, JUnit test codes, other required files (such as image and data files), and a README that explains how to run your software.
 - [PRJ#01-GroupNN-demovideo.mp4](#): (or a link to some other location): You should shoot a 3-5 min video where you as a group present all the functions and capabilities provided by your software.
 - [PRJ#01-GroupNN-testvideo.mp4](#): (or a link to some other location): You should shoot a 5 min video where you as a group introduce your source and test codes, you compile and execute your program to prove that it compiles and runs without error, and run your test suites to show the test coverages of your units.

Vantage Pro2 Sensor Suite

The Integrated Sensor Suite combines our rain collector, temperature and humidity sensors, anemometer, UV, and solar radiation sensors into one package. You can customize your Vantage Pro2 Plus to suit your needs and microclimate.

Click on the boxes below for more information.

1. Wind Direction	2. Wind Speed
3. Rain Collector	4. Built-in Bubble Level
5. Radiation Shield	6. Temperature/Relative Humidity
7. Solar Radiation and UV Sensors	8. Wireless/Cabled
9. Weather Proof Housing	10. Solar Panel
11. Easy Installation	



I. Problem Definition:

You are going to develop the core software for the [Wireless Vantage Pro2 Integrated Sensor Suite \(ISS\)](#) (Product number: 6322) manufactured by [Davis Instruments](#).

The ISS software shall

- collect data measurements from all attached sensors of the device according to the device specifications (provided on the website),
- process the sensor data properly to serve (or send) it to the data monitoring devices. These include [Wireless Vantage Pro2 Console Receivers](#) (Product number: 6316) and [Wireless Weather Envoy](#) (Product number: 6316) that transfers the incoming data to a stand-alone weather monitoring application running on your computer.
- transfer the data between the weather station and receiver or envoy devices on a (wireless) IP network by serializing the data.



You should develop the essential ISS software in Java and its unit tests using JUnit 5 to test your ISS software with a test coverage of 90% or above.

Since you do not have the actual sensors that can provide data to the ISS, you will also need to implement proxy sensors by following their specifications. For the sake of this assignment, you should implement all sensors collectively as “a single stand-alone application” or “a single driver class running on its own thread” separate from your ISS software and interchanging data with the ISS by serialized sensor and control data.

Hint: The textbook includes a case study on a wilderness weather station where you can benefit while you work on this project. Please see Chapters 5, 6, and 7 for relevant information.

II. Programming Guidelines:

1. You *should* include Javadoc comments at the beginning of each file, class with some necessary descriptive information, as well as some comments to the complicated parts of your code.
2. Please try to eliminate as many of the warnings displayed by Eclipse, and optionally other Eclipse plug-ins (like *CheckStyle*, *FindBugs*, and *PMD*) to improve the quality of your codes before submitting them.
3. You should code defensively. You should test setters and constructors for invalid values, as well as null values and thrown exceptions, or handle the error cases properly so that your program will not crash during its execution. //