CSC 454/554 Homework #2

I enjoyed our first lecture. Next Saturday, we'll focus primarily on OO Analysis… with a design pattern or two thrown in for fun.
Enjoy this week's reading and OO design problem!

Reading

Please read the following chapters of Larman's "Applying UML and Patterns":

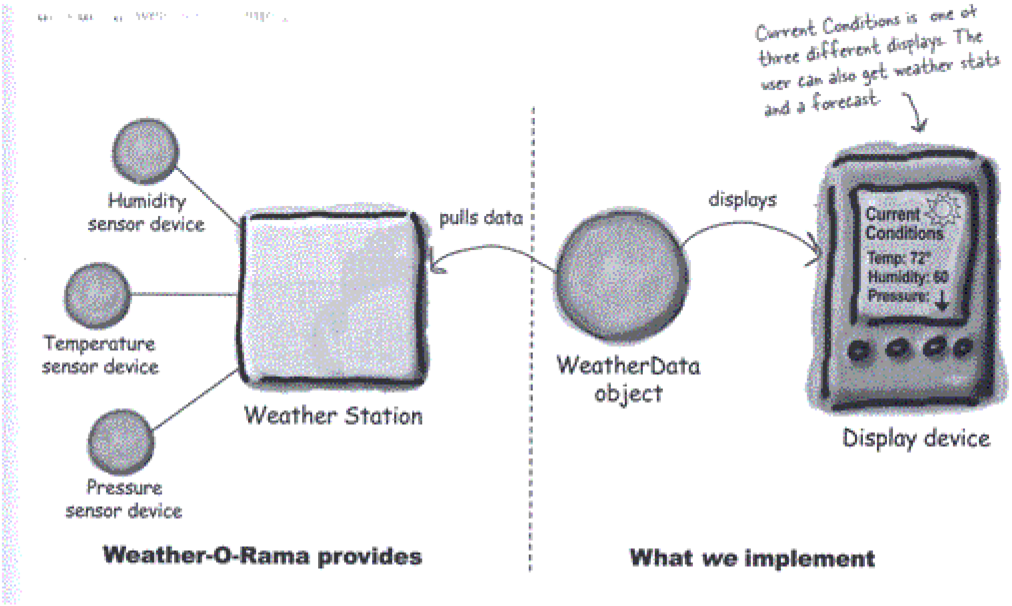
- Chapter 1 "OO Analysis and Design"
- Chapter 3 "Case Studies"
- Chapters 4-7… the "Inception" chapters discuss software requirements and their analysis.

From "Design Patterns Explained", please read:

- Chapter 18 "The Observer Pattern"

OO Design Problem

You've been hired by that dynamic, young startup Weather-O-Rama. They have a weather station all built and your team is to design the software to display the results. It'll work something like this:



Don't worry about how the `WeatherData` object will get stuff from the "Weather Station," rather focus on how other (display) classes will use `WeatherData` to access this data.

The `WeatherData` object should be usable for many different displays and/or display types. For the heck of it, assume that you have three different displays to worry about initially:

- `CurrentConditionsDisplay`
- `StatisticsDisplay`, and
- `ForecastDisplay`

We don't really need to know much about these displays other than they use `WeatherData` to do what they do. The number and types of displays is definitely an area of potential future change that you'll want to accommodate.

So, please:

1. Draw UML showing the `WeatherData` and 3 display classes. Also, what methods will `WeatherData` include to get it's data from the Weather-O-Rama equipment.

What else can I tell you…

- Um, does your `WeatherData` object **push** data out to displays or do displays **pull** it out of the `WeatherData`. Another term for this data (humidity, temperature, and pressure) is "state." You can try both and see which you like.
- Don't worry about the specifics of the display objects… graphics and such. For our (pretend) purposes, you could just have them print messages to the console.
- Also, please be ready to explain how additional displays can be added to your design.
- If it makes you warm and fuzzy, you can doodle a snippet of code, but it's not necessary. Sometimes it helps if you do this though.

2. Please create C++ or Java classes for your UML diagram. Hopefully, this step will be easier after last Saturday's lecture.

My email is wtkrieger@noctrl.edu
good luck… yow, bill

Created: September 17, 2006