Preface: The Point, or What Busy Meteorologists Do All Day

Virtually every job in meteorology and the atmospheric sciences involves the use of computers. This is primarily because meteorology is a discipline that requires the analysis of large amounts of data, and handling large amounts of data is what computers do best. From processing observations to processing model output, from producing analyses of remote sensing data to distributing weather information over the internet, meteorologists work with a variety of highly-specialized computer programs every day.

A number of software packages that decode, analyze, compute or contour weather data have been written. While some of these programs work on Windows PCs, most of them use computers that use the Unix operating system. Therefore, the familiar world of Microsoft software has to be left be left behind in this course. This is part of the reason that for this course we will also be reading In the Beginning… Was the Command Line by Neal Stephenson. Stephenson discusses the roles of various operating systems in different fields of study.

Not Another Computer Science Course

It is important to understand that this is not going to be a "computer science" course. In fact, we're going to deliberately do a lot of things differently than they would in a computer science course! Computer science courses are usually designed to expose the student to as many different features or capabilities of a computer language as possible, typically without worrying just too much about what you would actually *use* these capabilities for. This course is all about the application of skills to solving common problems in meteorology. We are actually going to learn surprisingly few C functions, yet we are going to produce sophisticated programs that work very much in the same way expensive software packages do. The emphasis of the course, then, is really more about "problem solving for meteorologists" than anything specifically about the C language.

Gardening

At some point, most parents get the idea that their children should try to raise a vegetable garden in the backyard. So all of the grass in one corner of the yard gets torn out, and the kids plant some peas, potatoes, onions, and tomatoes. All summer long, the kids have to water the plants, fertilize them, and chop out the weeds. At the end of the season, maybe the vegetables that are produced need to be canned, so someone has to go out and buy jars and lids and a canner. In the end, after all of that work, the kids can see the product of all of their hard work.

But sometime along the way, the kids will almost certainly have noticed that a ten-pound bag of potatoes costs $1.99 at the grocery store, green beans are three cans for a dollar on sale, and a bag of frozen peas is $1.49. On some hot day when the kids don't want to hoe in the garden, there is usually a battle between the kids and the parents about the whole gardening project.

The kids are right--there is no way that gardening is "worth the work". Almost all produce costs less to buy at the store than it costs to raise it yourself. The parents, of course, already knew this (probably because *their* parents made *them* raise vegetables

when they were kids). If pressed, the parents will probably explain that the *process* of raising the vegetables was the important lesson in all of this, because the kids will have learned two important life lessons:

1. While food is cheap and easy to buy at the store, it is important to appreciate the huge amount of work that *someone* went to in order to produce that food. This sort of appreciation helps kids know not to waste food and to value the work of others.
2. While raising the vegetables was a lot of work, successfully harvesting *some* amount of food shows the kids that *they are capable of doing this*. In other words, it is a challenge that the kids can successfully complete, which is very empowering.

In this class, we are doing the computer-programming equivalent of gardening for ourselves. Every single program that you are going to write already exists as a piece of software that you can buy or a web site that you can visit. You are going to go to a lot of work creating results and maps that are widely available, and I know that there are going to be occasions when you are frustrated and wondering why you are doing this. My response is that you are learning the same important lessons that you would be learning by raising a garden:

1. While the internet is teeming with web sites that will provide you will maps and diagrams to analyze the weather, it is important to appreciate the enormous amount of work that went into the creation of this information. Someone somewhere wrote the programs that make the plots that you see--and the programs that decoded the observations, and the programs that distributed the data over the internet, and the programs that performed the necessary calculations on the data, etc. In an era of cheap and easy exchange of information, it is important to appreciate the work that is done "behind the scenes".
2. More importantly, I want you to realize that *you are capable of creating programs like these*. The weather software and weather web sites that you visit every day were created by meteorologists *just like you*. While creating a whole suite of programs that would be able to produce satellite images, model output, radar plots, different kinds of contoured data and thermodynamic diagrams would require a lot of time, all of the techniques that would be needed are things that you are going to learn in this course. And while you will probably never again write programs that produce the exact kinds of output that you will produce in this course, over the years in your career as a meteorologist you are going to do *other* great things that are going to require *other* impressive computer programs.

If you still aren't sure that this is important, go to the class syllabus and read the section called "Goals".