24-623 Computer Logistics

A significant portion of your evaluation in this course will be based on your ability to develop computer programs, run them, and analyze the results.

You must write your own code. Using a software package is one thing. Developing your own code is another, and this experience will give you significant advantages in the future. When you submit your homework, you will also be sometimes required to submit your code.

You are free to write programs in any language you like. Options include C++ and FORTRAN. I have written code in both C++ and FORTRAN, and will be best able to assist people using those languages. In-class demonstrations will be done in C++ (I use a C++ compiler, but write code more typical of C). Code provided for some homework assignments will be in C++. **Using C++ is the best option.**

If you don't have FORTRAN or C++ available, I recommend using Crimson Editor for editing your code (free at http://www.crimsoneditor.com/) and installing the cygwin linux emulator (free at http://www.cygwin.com/ - download the entire package to make sure that everything will work, this may take a while) to get FORTRAN and C++ compilers. There may also be resources available on the CMU or ME department clusters.

If you don't know what language to use, choose C++. I will be able to best help you with that. There is lots of information about C++ on the web. A Google search on "C++ tutorial" yielded lots of hits, including:

- http://www.cplusplus.com/doc/tutorial/
- http://www.cprogramming.com/tutorial.html
- http://cplus.about.com/od/beginnerctutorial/l/blcplustut.htm

Work through these to get a grasp of the general concepts of programming. Your code will not be overly complicated. You will need to use *if* statements, *for* and *while* loops, and input and output data. Topics such as pointers and object oriented programming are beyond what you will need.

I will hold a coding workshop in the first week of class for interested students.

COMMENT YOUR CODE AS YOU WORK. Leaving yourself reminders about why something was done in a certain way will save you lots of time when you come back to your code, even after a week of not looking at it. Most people don't comment well. Start doing it from the beginning to make it a habit.

You may want to use math software packages at different times in the semester. You can get Matlab and Mathematica free from CMU by going to
http://www.cmu.edu/computing/software/all/index.html and following the links under software. For analysis, a spreadsheet/graphing program (e.g., Excel) will be very useful.

Please note the following if you plan to use cygwin as your C++ compiler.

1. When installing cygwin, you may need to run the setup.exe file twice. The following is from the cygwin website:
***
Run setup.exe any time you want to update or install a cygwin package.

Note that, when installing packages for the first time, setup.exe does not install every package. Only the minimal base packages from the cygwin distribution are installed by default. Clicking on categories and packages in the setup.exe package installation screen will provide you with the ability to control what is installed or updated. Clicking on the "Default" field next to the "All" category will provide you with the opportunity to install every Cygwin package. Be advised that this will download and install hundreds of megabytes to your computer. The best plan is probably to click on individual categories and install either entire categories or packages from the categories themselves.

Once you've installed your desired subset of the Cygwin distribution, setup.exe will remember what you selected so rerunning the program will update your system with any new package releases.
***

2. Your home directory is "C:/Program Files/cygwin/home".

3. Sample commands:
a. Create a directory with "mkdir"
b. Enter a directory with "cd"
c. Leave a directory with "cd .."
d. Compile the program test.cpp and create the executable test.exe with "g++ test.cpp -o test"
e. Run the program test with "./test"