

CMSI 585
PROGRAMMING LANGUAGES
(GRADUATE LEVEL)
Fall 2005

Assignment 0913

This is the “dive right in” assignment — it is meant to kickstart you into a “multilingual” frame of mind, and also to get you to set up the environment in which you will write your prospectus and research paper, if you haven’t already done so.

Not for Submission

1. Read Chapter 1 in Scott.
2. Get LaTeX up and running on whatever system(s) you will use to work on your paper.
3. Read Ray Toal’s “Introduction to...” Web pages, particularly for C, C++, Java, JavaScript, ML, and Perl: <http://www.technocage.com/~ray/classes/pl>
4. Get your hands dirty with the programming languages discussed in today’s class: setup working C, C++, Java, JavaScript, Perl, and ML development environments for yourself, type in the sample programs, and run them.

For Submission

Please submit all assignments on hardcopy; this is what I will count as proof-of-assignment. In addition, e-mail the source code of programming assignments to me. Again, the ground rule is: everything on hardcopy, and source code via e-mail also.

1. Type, customize, and print *paper.tex* and *prospectus.tex*, and submit the printouts to me (no problem, because you did #2 above, right?)
2. Choose a favorite algorithm similar in complexity (er, simplicity?) to greatest common denominator and implement it in C, C++, Java, JavaScript, Perl, and ML. For example, least common multiple, manual modulo, etc. Don’t get too fancy — the point of this assignment is to get hands-on with a programming language, and not the algorithm per se. Consider this to be a “learning by example” exercise, with some practice looking things up on the Web if you get stuck on a language point.

As stated in the syllabus, you presumably already know at least one of these languages well enough to write something pretty much in your sleep. In addition, many of these languages are syntactically very similar, so working via *copy-paste-tweak* is reasonable and acceptable.

Extra Credit (Sort of)

Write all of your programs as unit tests — that is, with the algorithm factored out cleanly, and using a unit test framework to express and run test fixtures. You may adopt the unit test frameworks or techniques in the sample code, find something else (but make sure I can run it!), or roll your own. The unit test frameworks used by the sample code, except for ML, are listed in the course Web site. The ML sample code embeds its own unit test functions; however, the overall approach remains the same.

P.S. This task is extra credit now only due to the time allotted. You *will* be required to do this eventually, in a future homework. The extra credit is for those of you who are able to do it by the due date of *this* assignment.