

CMSI 585
PROGRAMMING LANGUAGES
(GRADUATE LEVEL)
Fall 2005

Assignment 1129

We're on the home stretch, so this will be the last programming assignment for the semester. After this, it's paper, paper, paper. Don't forget to send me drafts of your paper as your work progresses.

Not for Submission

1. Read Scott Chapter 8.
2. Work on your paper!

For Submission

This last assignment is more exploratory, so we won't apply the old unit test format to this. However, the approach is highly recommended. As usual, submit code on hardcopy and by e-mail, and submit all other exercise responses on hardcopy.

You are asked to explore the individual subroutine capabilities of our 6 languages (C, C++, Java, JavaScript, ML, Perl) by implementing variations of the *twice* function. The *twice* function is supposed to take another function and a single argument, and is supposed to compose that function with itself. It is most concisely stated in ML:

```
fun twice f x = f (f x);
```

With this in mind, try the following:

1. Try to implement this "basic" *twice* function in all 6 languages. Note, YMMV, because the languages vary in terms of their subroutine support and design. So, you will not be able to do *twice* in exactly the same way in all languages.
2. Try to implement *twice* using a variable number of functions. Each function would be executed in sequence, twice. Again, YMMV, as each language has varying (or non-existent) support for variable arguments.
3. Write a summary of your attempts to implement *twice* in these ways for each language: discuss what can and can't be done, any limitations or constraints that you encountered, or anything else of interest that distinguishes the languages from each other.

Again, this is more exploratory, so the point isn't necessarily to write a "great" *twice* function, but to use it as a way to explore the subroutine features of the 6 languages.