Language Project

As a long-term project in this class, you will learn a new language and present it to the class. It must be a language that you do not know, that is not selected by anyone else in the class, and that has a free (preferably open-source) compiler so it can run on our machines. Here are some suggestions:

• Ada: Designed for government contracts. Puts an emphasis on correctness.
• Algol: Historic language that is pre-cursor to most of the languages you know (Java, C, etc).
• Chapel: Designed to support parallel programming, particularly for high-performance computing. Under fairly active development.
• Cobol: Historic language designed for business applications.
• C#: Language similar to Java designed by Microsoft as part of .Net framework.
• D: Language under active development. Similar to Java, but with programmer productivity as a specific design goal.
• Eiffel: Object-oriented language specifically promoting software design principles.
• Erlang: Designed for telecommunications switches, it is built for high-concurrency and fault tolerance.
• Forth: Stack-based language, significantly different from other imperative languages.
• Fortran: One of the first programming language, but still used within its targetted domain (numerical and scientific applications).
• F#: Functional language originally designed by Microsoft.
• Go: C-like language (by one of its creators) created at Google.
• Habanero Java: Now a Java library, but originally an extension of the language. Designed to support parallel programming.
• occam-π: Language based on formalism for mobile/distributed programming.
• OpenCL: Open-source language intended to allow programming of CPUs, graphics cards (GPUs), and other heterogeneous computing systems.
• Pascal: Imperative language designed for CS education. (Primarily popular in the 80s and early 90s.)
• Perl: Language focused on text processing.
• Postscript: Language used to represent images for printers.
• Prolog: Logic language designed for AI applications.
• Python: Probably the most popular scripting language at the moment.
• Ruby: Language designed for programmer productivity and fun (!?).
• Scala: Has both functional and object-oriented features. Runs as Java byte code so it has access to the Java libraries.
• Smalltalk: Influential object-oriented language. Literally everything is an object (even integers).
• SNOBOL: Historical text-oriented language.
Below are the parts of this assignment. All due dates are at 11:59pm on the indicated day except for the presentation, which obviously must be ready for class.

This assignment can be completed individually or (preferably) by pairs of students. Teams are expected to complete all the parts in a collaborative way. At the time of your last submission, each team member should separately email me talking about how you worked together.

I’ve reserved 11/2, 11/3, 11/4, 11/6, 11/9, 11/10, and 11/11 as presentation days.

1. (5 points) Submit the name of your language, where I can get the compiler (if it’s not already installed on euclid), 2–3 resources that you will use to learn about the language, and when you’d like to present. Language requests will be granted on a first-come first-served basis; you are encouraged to talk with your classmates so that you pick a language no one else is also choosing. Also, talk with me ahead of time if your language is not on the list above. **DUE: Friday October 16**

2. (10 points) Submit a description of the (at least) three sample programs that you will write using your language. These programs should be well-suited to its features (implying that you’ve at least started your research by now...), but not sample programs appearing in any of the resources that you are using. Include a description of the language features that you will illustrate with each of the programs. **DUE: Monday October 26**

3. (20 points) Present your language to the class. Your presentation should run for most or all of the period and include showing at least parts of your sample programs. You should also briefly put the language into context: when was it designed, by whom, and for what purpose? What features of earlier languages does it adopt and/or which of its features are incorporated into later languages?

   With your presentation, submit (to me) a potential homework problem for the class on your language. This should be a short program to write or modify using your language; remember to make these tasks simple enough for students with a single period of instruction in your language. **DUE: On the day of your presentation...**

4. (20 points) Submit your sample programs (include compilation and use instructions) and a 5-page writeup of your language. The content should follow the suggestions for the presentation. Include appropriate references. **DUE: Wednesday Nov 11**