Language Comparison Assignment
CS560 Winter 2005

Due Dates: Part 1: Jan 30th
Part 2: March 6th

Part 1) Pick three languages and describe the program you intend to implement. Turn in a sample run of a small program for each language.

Part 2) Turn in source code for each program, example runs illustrating that your program works, and a write-up comparing the languages.

Objective: To be exposed to the language features of several programming languages, and be able to discuss the differences between language features in several languages.

For this assignment, your task will be to explore three programming languages.

Requirements:

Picking languages: one language you can be an expert in, but the other two you should not be as familiar. For instance, maybe a Java programmer would select Java, C, and Python, where you have never used Python, and only used C for a few small programs.

Potential languages: C, C++, Java, Python, Perl, Ruby, Haskell, Lisp, ML, Scheme, C#, Ada, Fortran, Tcl, Javascript. These are just suggestions feel free to find others. Make sure there is a robust interpreter or compiler for the language.

Implementation: Choose a program to implement. I would suggest a game, or some other utility. If you have implemented a program in a previous class you can use that program, however, you need to implement the program in the other two languages. The program should be around 3000 LOC for an imperative/object-oriented language. I would like the program to be large enough that you are exposed to several features of the language. The length of the program is not critical, but you need to write some code to be able to write up an analysis of the languages.

Analysis: The final part of this assignment is to compare the different languages. The following is a list of topics that should be included in the analysis:

name bindings and scope rules (static, dynamic, modules, nested vs. non)
type system
- basic types
- type constructors (user-defined types?)
- other characterisitics (weak, strong, static, dynamic, explicit, inference)

memory management
- static (globals, code, class-data)
- dynamic (stack, heap, garbage collection, explicit)

call mechanisms (by-val, by-ref, by-val-res, by-name, etc...)
eval strategy (eager, lazy)
refs/pointers as first-class types higher-order functions
functions as first-class types object/class abstraction
inheritance (single, multiple)
nondeterminism
polymorphism (templates, generics, functors)
concurrency (processes, coroutines, threads, fork/join)
compiled vs. interpreted vs. mixed
main intended use or domain of application
support for or use of assertions/invariants/pre-post-conditions/contract

In addition, you should provide a performance and size comparison of the languages. Time for compilation/interpretation, size of executable, runtime.

The reference below provides a good comparison between languages. It is specific to object-oriented languages, but it a good example. Your write-up should be 8-10 pages.

References:

R. Henderson and B Zorn. A Comparison of Object-oriented Programming in Four Modern Languages SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 24(11), 1077–1095 (NOVEMBER 1994).