Homework 10

Due Thursday, 4/18/2019

Please turn in your homework solutions as usual at `https://submit.cs.pomona.edu/2019sp/cs131`. There is no coding in this assignment, so you do not need a separate Haskell file.

1. (15 points) **Subtyping and Visibility**

   Please do problem 12.6 from Mitchell, page 375.

2. (10 points) **Subtyping and Specifications**

   Please do problem 12.9 from Mitchell, page 377.

3. (10 points) **Subtyping and Binary Methods**

   Please do problem 11.8 from Mitchell, page 334.

4. (15 points) **C++**

   The following is an excerpt from "An Overview of C++," by Bjarne Stroustrup, SIGPLAN Notices, 1986-10:

   "Section 6. What is Missing?

   C++ was designed under severe constraints of compatibility, internal consistency, and efficiency: no feature was included that

   (a) would cause a serious incompatibility with C at the source or linker levels.

   (b) would cause run-time or space overheads for a program that did not use it.

   (c) would increase run-time or space requirements for a C program.

   (d) would significantly increase the compile time compared with C.

   (e) could only be implemented by making requirements of the programming environment (linker, loader, etc.) that could not be simply and efficiently implemented in a traditional C programming environment.

   Features that might have been provided but weren't because of these criteria include garbage collection, parameterized classes, exceptions, multiple inheritance, support for concurrency, and integration of the language with a programming environment. Not all of these possible extensions would actually be appropriate for C++, and unless great constraint is exercised when selecting and designing features for a language, a large, unwieldy, and inefficient mess will result. The severe constraints on the design of C++ have probably been beneficial and will continue to guide the evolution of C++."

   Given this description, please contrast the design goals to those of Smalltalk. What is different? What is the same? What is the intended use of C++, and how does that influence the design?

   You do not need to read Stroustrup's paper to answer this question, though you might find it interesting. If you want to read it, you can find it at `http://www.cs.pomona.edu/classes/cs131/readings/stroustrup.pdf`. *Note that this paper is from the very early days of C++, so features like parameterized classes and multiple inheritance have since been added.*