Homework 10

Due Friday, 4/15/11

Please turn in your homework solutions online at `http://www.dci.pomona.edu/tools-bin/cs131upload.php` before the beginning of class on the due date.

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. (25 points) **"Growing a Language" Reading**

   Please read the short paper entitled "Growing a Language" from the auxiliary reading page. The author, Guy Steele, is one of the designers of Java, and this was his keynote talk at the major American conference on object-oriented languages (OOPSLA) in 1998.

   This paper includes a discussion of why some items were originally left out of Java when they designed the language, but likely should have been included. It also gives some insight into programming language design in general. The following questions will guide you through the paper:

   (a) What is wrong with designing/using a small language like Lisp?
   (b) What is wrong with designing/using a huge language (C++)?
   (c) What was Steele's goal in designing Java?
   (d) Why does Steele feel that overloaded operators are important. Contrast this with his conviction (expressed elsewhere) that overloaded methods might have been left out without much harm.
   (e) Discuss why generics might have been left out originally, but why he now feels they are important.

   Write a separate paragraph discussing each of these items. A few sentences on each part is sufficient.

   We will spend some time discussing these points after the homeworks are turned in.

4. (15 points) **Smalltalk Reading**

   Read "Design Principles Behind Smalltalk" by Dan Ingalls (available on the "links" page).

   Think about the Principles set forth. Which are new? Which have we seen before? What principles are remniscent of Lisp? What is the scope of the language (ie, what is included in its definition)? A few sentences on each item is sufficient— clarity is more important than length of answer.

5. (15 points) **C++**

   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++."

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 search for it through the ACM digital library, `http://portal.acm.org/`.