Homework 6: OO Languages and Parameterization

Due: problem 1, Tuesday, April 30; the rest on Thursday, May 2.

In this homework, you will learn more about OO programming languages.

We will also investigate the parameterization principle, working from Schmidt's book, chapter 3. Since there isn't much time to "lecture" on this material, we will take it as an opportunity to have you be more self-taught. In other words, read the material in chapter 3 on your own. (For all the problems from Schmidt's book, use standard notation.)

Unless you choose problem 9, this is an individual homework; all of problems 1–8 are to be done on your own.

You may substitutue problem 9 for problems 2–8, and you may do problem 9 as a group problem.

1. This is a reading problem. Find a published article about: the design of an object-oriented programming language, type systems for object-oriented programming languages, or the semantics of object-oriented languages. Note that an article about "how to program" in an OO language is *not* acceptable.

   Read the article and summarize it, writing something for each of the following parts. (Don't make this very long, please. You have a maximum of two (2) pages.)

   (a) (20 points) What is the problem(s) addressed by the paper? (If you read a survey paper instead of a paper solving a problem, then say that and describe instead the areas covered by the survey.) Do you think it is a significant problem (or, for a survey, that these are significant areas)? Why?

   (b) (40 points) Describe the main points of the argument and the solution(s). (For a survey paper, describe the main techniques surveyed.) Try to relate this to the material we have studied earlier in the class, in particular to the type rules and soundness if you read a paper about OO type systems. If you read a paper about a language or semantics, what kind of abstractions and parameter mechanisms are there and how do they compare to what we have studied so far?

   (c) (20 points) How is this claimed to be an advance on related work? (For a survey paper, this point is moot unless certain ideas are claimed as new with that paper. If there are no such claims, then just say so.) Do you believe these claims? Why?

2. (suggested practice) Do exercises 1.1, 2.1, 3.1, 3.2, 3.4(a), on pages 98–100 of Schmidt's book.

3. (40 points) Do exercise 4.3(a) on page 100 of Schmidt's book.

4. (10 points) Do exercise 5.2 on page 101 of Schmidt's book.

5. (suggested practice) Do exercise 5.3 on page 101 of Schmidt's book. (You are to just give the syntax changes and the typing rules.)

6. (suggested practice) Do exercise 7.1 on pages 101–102 of Schmidt's book, parts a, b, and c.

7. (60 points) Do exercise 7.2 on page 102 of Schmidt's book, parts a, b, and c. When showing that mechanisms are different, give an example program and state what the output is using each mechanism. (You don't have to explicitly calculate the result, although that would help you debug your semantics.)

8. (suggested practice) Do problem 10.2 on page 103–104 of Schmidt's book.

9. (points = the total of problems 2–8 above) You can do this instead of problems 2–8 above, either individually or in a group (I suggest doing it as a group). Implement an interpreter for the language of Schmidt's chapter 2, without recursive abstractions, but with parameterization as described in Schmidt's chapter 3. Do this using Smalltalk, and OO techniques. Your grade will be based on how clear your design is, on how sensible your added parameterizations are, and how much of chapter 3 ideas you cover. Include type rules and semantics in Schmidt's notation for all the parts of the language you changed from chapter 2 as documentation.