Homework 4: The Abstraction Principle

Due: all individual problems on March 21; all group problems due on April 2.

In this homework you will learn about the abstraction principle, including module systems and records.

All problems except for problem 17 and the ones following it are individual problems. Only problem 17 and the ones following it are group problems. Please make a careful distinction, as it is cheating to share answers or code for individual problems.

For all Haskell programs, you must run your code with a Haskell interpreter (such as `hugs`). You must also provide evidence that your program is correct (for example, test cases). Hand in a printout of your code and your testing. Some questions just require an answer in English. Be sure to clearly label what problem each function solves with a comment.

Read Schmidt's book, chapter 2.

1. (suggested practice) Do problems 1.1, and 1.2 on page 67 of Schmidt's book.
2. (10 points) Do problem 2.1 on page 67 of Schmidt's book. This is to be done in Schmidt's notation. Your answer can be handwritten, although you may find it easier to manipulate this kind of calculation from inside a text editor. (Hint: be careful to only do the *well-formed* programs from exercise 1.1.)
3. (20 points) Do problem 2.2, part a on page 68 of Schmidt's book. This is to be done in Schmidt's notation. Your answer can be handwritten.
4. (suggested practice) Do problem 2.2, part b and c on page 68 of Schmidt's book.
5. (20 points, extra credit) Do problem 6.3 on page 68 of Schmidt's book.
6. (10 points) Do problem 4.1 on page 68 of Schmidt's book, but only for part (b) of exercise 1.1.
7. (20 points, extra credit) Do problem 4.3 on pages 68 of Schmidt's book.
8. (suggested practice) Do problem 5.1 on page 68 of Schmidt's book.
9. (20 points, extra credit) Do problem 6.2 on page 68 of Schmidt's book.
10. (60 points) Do problem 7.2 on page 69 of Schmidt's book. However, do the typing rules and semantics in Haskell notation. Use the following abstract syntax in Haskell notation as the start of your work.

```
> type K = Completion
> type C = Command
> type E = Expression
> type L = Location
> type N = Numeral

> data Completion = C `KSemi` K | Stop
> data Command = L `Assign` E | C `Semi` C | While E C | Complete K
> data Expression = Num N | Deref L | E `Plus` E | Not E | E `Equals` E
> data Location = Loc Integer
> type Numeral = Integer
```

You should also be able to build on the type checking code for chapter 1. Be sure to test your code; hand in a printout of your testing along with the code.

11. (suggested practice) Do problem 7.3 on page 69 of Schmidt's book.

12. (40 points, extra credit) Do problem 8.2 on page 69 of Schmidt's book.

13. (40 points, extra credit) Do problem 8.3 on pages 69-70 of Schmidt's book.

14. (40 points, extra credit) Do problem 8.4 on page 70 of Schmidt's book.

15. (30 points) Do problem 11.1 on page 70 of Schmidt's book. This can be handwritten. Try to relate your answer to the semantics and be precise.

16. (50 points, extra credit) Do problem 11.2 on pages 70-71 of Schmidt's book.

17. (100 points) This is a group problem. Code the the semantics for the language of chapter 2 in Haskell. Work from your code for the language of chapter 1, and section 2.15 in Schmidt's book. Then on this base, work (a) problem 10.2 on page 70 and (b) problem 12.3 on page 71 of Schmidt's book. Do both using Haskell notation. Be sure to test your code. (Use problem 13.1 as some test cases.)

18. (50 points, extra credit) This is a group problem. Use a monad to write your semantics for the previous problem.

19. (40 points) This is also a group problem. Do either exercise 13.4 or 13.5 on pages 72-73 of Schmidt's book. Give your answer to the problem you selected, and also extend your Haskell semantics from problem 17 to work with the new language features. Be sure to test your code.

20. (50 points, extra credit) This can be either done as a group problem or individually. Do problem 14.1 on page 73 of Schmidt's book.

21. (50 points, extra credit) This can be either done as a group problem or individually. Do problem 15.1 on page 73 of Schmidt's book.