Homework 6: Dynamic Semantics

This homework is due Friday, April 26.

1. (15 points) Consider the following Scheme procedure:

```
(define cycle-proc
  (lambda (th)
    (letrec ((loop (lambda () (th) (loop))))
      (loop))))
```

   (This is program 14.11 in *Scheme and the Art of Programming.*) `cycle-proc` takes a thunk (a parameterless procedure), and executes it indefinitely. Your problem is to write `cycle-proc` using `call/cc` instead of recursion.

2. (extra credit only) Read section 17.6 of *Scheme and the Art of Programming* and do one or more of exercises 17.10, 17.11, or 17.12 on page 571. You may wish to do some of the earlier exercises in chapters l16 and 17.

3. (20 points) Do problem 4 on page 214 of Schmidt's text (continuation semantics and weakest preconditions).

4. (extra credit only) Do exercise 7 on page 215 of Schmidt's text (see chapter 8 for the principles).

5. (10 points) Do problem 13, part (a), on page 216 of Schmidt's text (translating direct to continuation semantics).

6. (a) (20 points) Give a standard semantics (with continuations) for the language PS(**Lif**) based on Felleisen's figure 3 and the LISP-if (**Lif**) described at the bottom of page 21.

   (b) (5 points) If the predicate `int?` was included in the language PS(**Lif**), what would it's denotation be? (That is, give the denotation of `int?`.)

   (c) (extra credit only) Can you argue from your semantics why `int?` cannot be macro-expressed in PS(**Lif**)? Why or why not? (Hint: it may be helpful to read Felleisen's remarks about "full abstraction" on page 34, and Schmidt's discussion of full abstraction.)