P.F. Reynolds, Jr. **17 March 1998**

CS655 Midterm

This is a two page exam with two parts. Answer *five* of the seven questions in Part I and *one* of the two questions in Part II. Questions in Part I make up 75% of your point total (15 points each); the question from Part II makes up the other 25%. Submit your answers on separate 8-1/2 by 11 sheets of paper. Start a new page for each answer and put answers on only one side of a page. Limit answers in Part I to one page each and answer in Part II to two pages. Put your name at the top of each page. Identify your answers with the five-character mnemonics associated with the questions. *Submit your answers stapled and in lexicographic order.* Type your answers. This exam is open books, *your* notes, etc. However, you are to speak to no one other than me or John Viega about it. You must stop answering questions on this exam within 15 hours after you first read any part of it, or any posting about its contents. Turn your exam in to the dept secretaries as soon after your 15 hours are up as CS office business hours will permit, but no later than 5PM ET, Friday, 20 March. Late exams accepted by prior arrangement only.

Pledge your exam: I have neither given nor received unauthorized aid on this exam, I have abided by the 15 hour rule, and I have not been a party to the removal of materials from the CS or Sci/Tech libraries to the detriment of others taking this exam.

Watch the CS655 homepage for clarifications and answers to your questions. A copy of this exam can be found on the CS655 homepage.

Part 1: answer five out of seven

1DEFA) In PL/I, references to undefined objects in an inner block are defined by default in the outermost encompassing block. a) Give an example where this is a desirable assumption. b) Give an example where this is an undesirable assumption. c) Give an example where having default declarations appear in the outermost block in which the reference appears is a better assumption. Explain and support your examples.

1LAMB) Write a lambda abstraction that takes a parameter *n*, and computes the n^{th} fibonacci number. Show what reductions will be performed when your abstraction is applied to the argument 4, and converted into normal form.

1ORRE) Give two examples for each of *three* of the five following languages, where the first example demonstrates how the language is orthogonal but not regular and the second demonstrates how the language is regular but not orthogonal. Make the nature of your (six) examples as different as you can. The languages: a) ALGOL60, b) ALGOL68, c) Pascal, d) Ada83, and e) Haskell. Explain each example.

1REDO) Recall the higher order filter function, FIL P, and how it could be used to correctly implement set operations on sequences. Consider the function, RED OP, which applies numerical reduction operations on sequences of integers and produces a single value, *acting as if the sequence were a set (no repeated elements).* For example, RED [+] (2 4 8 4) would have 14 as its value, and RED [*] (2 4 8 4) would have 64 as its value. In each case the repeated 4 is ignored since we're treating the sequence as a set. Do the following: a) define an archetype for RED OP ≡ S → and b) give a correct Haskell definition for the function (RED OP) S. Note: RED OP () = 0 ∀ OP and RED OP (4) = 4 ∀ OP, e.g. RED [+] () = RED [*] () = 0, and RED [+] (4) = RED [*] (4) = 4. Note RED OP is deterministic only when OP is commutative (or there are fewer than two arguments).

1ROTE) Give examples showing the relative power of routine texts (ALGOL68) as compared to a) call-by-name (ALGOL60) and b) function parameters (e.g. Pascal). Support and explain your examples.

1SEMA) Extend the denotational semantics of the lambda calculus presented in Peyton-Jones with equations for power, division and modulus operations, that are lazy where appropriate. Do not use mathematical operators for modulus or power in your answer (you may use division).

1SWAP) Consider the following procedures:

```
procedure swap(x,y: integer);
    procedure f() : integer;
      var z: integer;
    begin
     z := x;
     x := y;
     return z
    end f ;
begin
    y := f
end swap;

procedure incr() : integer;
begin
    i := i + 1;
    return i
end incr;
```

And the call:

```
swap(i, A[incr()])
```

Describe the effects of the call under the following parameter-passing methods:

a) call-by-value-result
b) call-by-reference
c) macro expansion (i.e., if swap(x,y) were a C-like macro)
d) call-by-name

Part 2: answer one out of two

2EXAS) It has become common for languages to treat assignments as expressions, rather than as statements. Compare and contrast the two approaches. Which choice do you think a language designer should make, and why?

2SIFR) Argue either that Haskell is completely side-effect free, or that it isn't.

Pledge your work:

On my honor as a student I have neither given nor received unauthorized aid on this exam, I have abided by the 15 hour limitation and I have not been a party to the removal of materials from the CS or Sci/Tech libraries to the detriment of others taking this exam.