P.F. Reynolds, Jr. **7 May 1999**

CS655 Final - Part II

This is part II of a two part exam. Answer three of the four questions in the 2XXXX section (10% each) and one of two questions in the 3XXXX section (20% of your grade). Submit answers on separate 81/2 by 11 sheets of paper. Start a new page for each answer and put answers on only one side of a page. Limit 2XXXX answers to one page each, and 3XXXX answer to two pages. Put your name at the top of each page. Identify answers with the five-character mnemonics associated with the questions. Submit answers typed, stapled and in lexicographic order. This exam is open books, your notes, etc. Communicate with no one other than Jim Gunderson or me about it. Stop answering questions on this part within twelve hours after you first read any part of it, or any posting about its contents. Turn this part of your exam in to CS dept secretary as soon after twelve hours are up as CS office business hours permit, but no later than 5PM ET, Tuesday, 11 May. Or, e-mail answers to (both!) pfr and gunders @virginia.edu. Format must be Word95/97 or PDF. If you don't receive an acknowledgement within 12 hours, e-mail answers again. Late exams accepted by prior arrangement only.

Pledge this part of your exam: I have neither given nor received unauthorized aid on this exam, I have abided by the twelve 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.

Answer three out of four from 2XXXX

2COIN) Give an example where (implicit) coercion and type inferencing interact in a way that can lead to unexpected results for the programmer. Suggest an improvement.

2DBYC) Give DBC-like REQUIRE, ENSURE, INVARIANT and VARIANT assertions for Scott's problem (loop odd(x) -> 3x+1; even(x) -> div 2 until 1). Discuss issues that may arise.

2ELEG) Defend the new MacLennan principle of elegance. How does elegance fit in with the other principles we've discussed? The definition of elegance is: Confine your attention to designs that look good because they are good.

2PAPO) Argue either that Java needs parametric polymorphism, or that there are reasonable alternatives in Java that preclude the need.

Answer one out of two from 3XXXX

3ORTH) You've just completed the design of your new programming language, OYSTER. Now ACM has come calling, asking you to write an article in defense of your language design philosophy. Write, for OYSTER, a concise equivalent of Larry Wall's article about Perl (Communications of the ACM, April 1999), addressing in particular your views on orthogonal language design.

3PROT) Protection of named entities is an important issue in language design. C++/Java have private, protected and public (plus C++ friends); Eiffel has secret/non-secret and finer control over name export. Describe your idea of an ideal approach to protection in languages that support subtyping, subclassing and specialization. Defend your approach.