P.F. Reynolds, Jr. **23 March 1999**

CS655 Midterm - Part II

This is part II of a two part exam. Answer *two* of the three questions in the 2XXXX section and *one* of two questions in the 3XXXX section. Questions in the 2XXXX section make up 30% of your point total (15% each); the 3XXXX question counts 25%. Submit 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 the 2XXXX section to one page each, and in the 3XXXX section 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 Jim Gunderson about it. You must stop answering questions on this part of the exam within <u>eight</u> <u>hours</u> 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 eight hours are up as CS office business hours will permit, but no later than 5PM ET, Friday, 26 March. 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 eight 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 *two* out of three from 2XXXX

2ASPC) How well do Kiczales' aspects satisfy: 1) the abstraction principle, 2) the localized cost principle, 3) the manifest interface principle and 4) the security principle? Explain your answers.

2PORT) Consider yourself a member of a language design team that proposes to pursue the portability principle for a feature rich language with the same fervor the ALGOL68 designers pursued orthogonality. Identify issues that must be addressed to meet the portability goal and for each issue you identify give an indication of how the portability goal might be met.

2PRED) Give three examples for each of *three* of the five following languages, that demonstrate that the language is not predictable. Make the nature of your (nine) examples as different as you can. The languages: a) ALGOL60, b) ALGOL68, c) Pascal, d) C, and e) Haskell. Explain each example.

Longer answer: Answer *one* out of two from 3XXXX

3FUNC) Advocates of the functional language paradigm argue that even if a pure functional language is not practical, the style of functional programming will still be the core of a good programming language. Do you agree? Why or why not? Support your answer with examples from ML and/or Haskell that you think help make your case.

3PRPR) A bleak assessment: 1) the greatest aid to programmer productivity is someone to answer the phone, 2) the C++ runtime is so complex it interferes with development of efficient and understandable code and 3) C and FORTRAN are still among the most widely used languages. Wow.

Identify three significant concepts in language design that showed great promise, but now appear to have had little impact on programmer productivity. Candidate concepts include: block structure, single entry - single exit control, modules and orthogonal language design; feel free to consider others. Defend your choices and explore why the concepts had little effect on productivity. Conclude with an analysis of what it may take in language design to introduce order of magnitude improvement in programmer productivity. (Finally, solve world peace -- no wait, save that for the final.)