

CS655 Final - Part I

This is part I of a two part exam. Answer five of the six questions in this part (50% of your point total; 10% each). Submit answers on separate 8 1/2 by 11 sheets of paper, each answer on a new page, one-sided. Limit answers to one page each. 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 five out of six

1ASEH) Identify and discuss issues that may arise regarding the interaction of aspects and exception handling. For one of the issues you identify, propose a defensible alternative.

1ITER) Explain the Seive! and Primes! iterators in Figure 1 of the Sather iterators paper. The paper makes reference to the "expressive power" of iterators in Sather. Make clear what that expressive power is in the Seive! / Primes! example.

1MUIN) Derived class *D*, which inherits multiply from base classes *B1* and *B2*, contains a reference to method *Ma* in *B1*. *Ma* contains an embedded reference to *Me*, which is defined in *B1*. However, *Me* is also defined in *B2*, and *B1.Ma*'s reference to *Me* gets bound (correctly) to *B2*'s *Me*. How can this happen? How do you suggest we change the language to prevent this? (Removing multiple inheritance is not an option.)

1REGA) Outline a design for free space reclamation that consists (solely) of a combination of reference counting and garbage collection. Make your objectives low cost, localized cost, and distributed cost.

1SUSI) Extend the Day/Liskov notion of subtypes to side-effects. (Day/Liskov address contra/co-variance of arguments, return types and exceptions. Add a fourth category: side-effects). Assume that all side-effects can be regarded as changes to non-local variables. Defend your chosen extension.

1TOON) Give a C++/Java program fragment that would be difficult or impossible to express in a Toon-talk kind of visual language. Explain the difficulty.