What’s on the final (2nd midterm)?

Schedule
- Drop box open soon
- Friday final (second midterm)
- Monday (8:30-10:30) project presentations — in this order — aim at 12 minutes each!
  1. c
  2. a
  3. h
  4. d
  5. e
  6. f
  7. b
  8. g

Ground rules – same as 1st midterm
- 50 minutes
- Open notes, open book
- Closed any kind of electronics (including calculators and cell phones)
- Closed邻席
- 5 questions, each 20 points, 100 points total

CSE403 Sp10

Question 1: program proofs
- [10 points] The following program (no loops in the program!) is annotated with some assertions (predicates), with some left as placeholders. Compute the most general predicates for the placeholders.
- [10 points] The following loop has a proposed loop invariant (I) that is intended to allow these three properties to be proven
  - \( P \Rightarrow I \) (the precondition implies the invariant)
  - \( \{B \land I\} S \{I\} \) (the loop maintains the invariant)
  - \( \{\neg B \land I\} \Rightarrow Q \) (termination implies the invariant)
  One of these three does not hold. Which one, and why?

CSE403 Sp10

Question 2: test coverage
- [8 points] Assess a given test suite in terms of one or more coverage dimensions (statement, edge, path) with respect to a given program
- [8 points] Create one or more tests that improve the coverage for the program
- [4 points] One short-answer question about coverage and infrequently called error handling code

CSE403 Sp10
Question 3: mutation testing

- [15 points] A question that gives a program, a test suite, and a set of mutations on the program, where you compute which mutants are killed/not-killed.
- [5 points] A short-answer question about the conditions needed for mutation testing to provide effective information about test suites.

Question 4: symbolic evaluation

- Given a program, perform symbolic evaluation over it, specifically
  - [10 points] Create a symbolic execution tree that explicitly tracks path conditions
  - [10 points] Solve the path conditions —“how do you get to this point in the execution tree?” — to define test inputs that cover all reachable statements.

Question 5: refactoring

- [20 points] Consider the following refactoring – a program and a refactored version of it. Under what conditions does it maintain equivalent behavior (that is, the original and the refactored program always transform program state the same way – performance and other dimensions can vary, though)?