Compilers [Fall 2015]
Practice Test III

NAME: _____

Instructions:

1) This test is 4 pages in length.

2) You have 2 hours to complete and turn in this test.

3) Prose-response questions include a guideline for how much write. Respond in complete English sentences. Essays should be well organized and readable.

4) This test is closed books, notes, papers, friends, neighbors, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.

6) Write and sign the following: "I pledge my Honor that I have not cheated, and will not cheat, on this test."

_____

_____

Signed: _____

1. [20 points]
What is a compiler? [1 sentence]

2. [20 points]
List the primary disadvantages of reference-counting garbage collectors.

3. [30 points]
For the following question, make all the DISM-memory-layout assumptions we made in class, ensure all the DISM-memory invariants we ensured in class, and respond in pseudocode.

What is an algorithm for correctly generating DISM-style machine code equivalent to the expression e_1 - e_2 ?

4. [30 points]
Why is it undecidable to determine whether non-garbage heap data can be collected? Provide a basic, high-level proof.

Undergraduates stop here. The remaining problem is for graduate students.

5. [25 points] [Essay]
A bootstrapped compiler C deterministically translates language S to language T and may be *malicious* (i.e., C may implement the backdoor-attack we discussed in class). Suppose we compile C with both C and *n* other, different compilers, all of which also deterministically translate S to T. At least one of these other *n* compilers is *benign* (i.e., does not insert backdoors); call the benign compiler C'. Describe the outputs of C(C) and C'(C) in *two cases*, first when C is benign and second when C is malicious. E.g., what languages are the outputs in, and are the outputs identical, functionally equivalent, or not even equivalent? Also describe the outputs of (C(C))(C) and (C'(C))(C) in those same two cases. Discuss what, if anything, we've accomplished by compiling C in this way.