YOUR NAME: _____

CSE401 Final (aka Second Midterm), December 10, 2008

- Open book, open note
- Closed electronics, closed neighbor
- 60 minutes (but I'll stay for 90 if anybody remains)
- Put your name clearly on top
- Legibility is a plus – if I can't read your answer, I won't try to read your mind

- *Don't open until you are told to*

1. (20 points total, 4 points/question) True or false – and briefly (about one sentence) explain your answer
    a. Generating target code directly from an annotated abstract syntax tree is feasible.

    b. An $O(N^2)$ algorithm highly optimized by a compiler generally runs more quickly than an unoptimized O(N log N) algorithm.

    c. A front-end based on an LR-parser places different demands on the back-end than would a front-end based on an LL-parser.

    d. A programming language with automatic memory management (e.g., garbage collection rather than alloc/free) simplifies programs but complicates the compiler.

    e. A compiler for a programming language with automatic memory management relies upon an intermediate language that supports automatic memory management.

2. (35 points) Some programming languages (such as C++, Eiffel, and Common Lisp/CLOS) support multiple inheritance: a class can be declared with multiple superclasses, inheriting methods and attributes from all such superclasses. For each of the following compiler tasks, explain how multiple inheritance complicates the task (if at all) and sketch (in 1-2 sentences only) an approach to handling such complications.

    a. (15 points) The data representation of instances of a class defined using multiple inheritance.

    b. (15 points) Invoking a method from an instance of a class defined using multiple inheritance.

    c. (5 points) Managing the equivalent of vtables.

3. (40 points total) A *stack machine* describes a computer architecture where instructions take no operands: most instructions, including load and store, implicitly operate on values at the top of the stack and replace those values with the result. (These arise both in actual processors, such as the Burroughs large system architecture, as well as in virtual processors, including the UCSD p-machine and the JVM instruction set.)

   a. (15 points total) From a compiler's point of view, concisely but clearly describe two distinct issues that would affect the Intermediate Code Generation phase due to the use of a stack machine as a target (instead of a register-based machine such as the x86).

   b. (15 points total) From a compiler's point of view, concisely but clearly describe two distinct issues that would affect the Target Code Generation phase due to the use of a stack machine as a target (instead of a register-based machine such as the x86).

   c. (5 points total) Describe one conventional optimization that would still be effective for a stack machine.

   d. (5 points total) Describe one conventional optimization that would no longer be effective for a stack machine.

4. (10 points) From a technical point of view, describe (in 1-2 sentences each) two things you learned about arrays from the MiniJava assignment.

5. (10 points) Concisely describe two kinds of information that can be computed during one execution of a program and then used to better optimize the program for further executions.

6. (10 points) The MiniJava compiler, like all compilers, selects an evaluation order for code generation: for example, at a node representing a binary arithmetic operation, is the invocation to generate the code for the left operand made before the one for the right operand, or vice versa. Question: Might the order affect the total number of registers required to evaluate the subtree? (And, briefly, why?)

7. (10 points) Consider the following code snippet:

```
if (little-endian) then
     S1;
else
     S2;
endif
…
if (little-endian) then
     S3;
else
     S4;
endif
```

Assume **`little-endian`** is intended to be a boolean "configuration" variable that selects whether **`little-endian`** or **`big-endian`** representations will be used in the program. In what way does that affect, if at all, potential optimizations?