

# Computer Engineering 175

## Project Guide

### 1 A Simple C Compiler

**What is Simple C?** Simple C is a subset of the C programming language. It has no user-defined types, few built-in types, a simplified declaration syntax, a streamlined set of operators, and only simple control constructs.

**Is every Simple C program a C program?** Yes, Simple C is a true subset of C, so every legal Simple C program is a legal C program. In particular, every Simple C program can be compiled using a traditional C compiler.

**Why use a subset of C?** Although Pascal is a smaller language and perhaps a better choice for building a first compiler, most students are familiar with the syntax and semantics of C, C++, or Java.

**Where will the compiler run?** You will be generating code for the Intel 80x86 family of processors running the Linux operating system. These machines are readily available in the design center.

**Why use an Intel platform?** Most students should be familiar with the Intel architecture through previous courses. Additionally, it is easy to write a simple compiler to generate efficient code for the Intel architecture. Generating efficient code for architectures such as the Sparc architecture would require a more sophisticated compiler.

### 2 Grading

**How will my compiler be graded?** Your compiler will be graded in six phases. Each phase will have a weight assigned. After each phase, you may continue with your implementation or use the solution provided.

**How will each assignment be graded?** Grading will be fully automated. Therefore, your compiler *must* produce output exactly as indicated in the assignment and online examples. Incorrectly formatted output that is otherwise correct will receive a zero. Your score will be determined by how many test cases your compiler passes. Your design and coding style will *not* be graded.

**How will I submit each assignment?** Assignments will be submitted online. The submission page is located off the class home page. Your username and password are the same as those of your design center account. A problem with the submission system is *not* a valid excuse for failing to complete an assignment.

**What do I submit?** For each assignment, you *must* submit a `tar` file containing your project directory, which *must* be named `phasen`, where `n` is the number of the current phase. Within the directory, you *must* have a `Makefile` that will produce an executable file called `scc`. The following steps will be used to compile your assignment:

1. `tar xf submission`
2. `cd phasen`
3. `rm -f *.o scc core`
4. `make`

**How will my compiler work?** Your compiler will read input from the standard input, will write valid output to the standard output, and will write error messages to the standard error: `scc < input-file > output-file 2> error-file`. Your compiler *must* work on the Linux machines in the design center.