Professor: Torgersen
Office Phone: 758-5536
Office Hours: Monday and Wednesday 2:00 to 4:00. Also by appointment.
Facilities: Grad lab, undergrad lab, laptops, telesto, freyr, and genesis.

Goals:

1. A basic understanding of compiler construction
   (a) Theory
      i. Finite Automata
      ii. Grammars
      iii. Attribute Grammars
   (b) Techniques and topics
      i. Lexical Analysis.
      ii. Symbol tables.
      iii. Parsing, and parse trees
         A. parsing algorithms, top-down vs. bottom-up parsing
         B. operator precedence parsing, recursive descent
         C. LR parsing, automatic construction of LR parsing tables
      iv. Syntax trees
      v. Semantic checking
      vi. Intermediate code
      vii. Introduction to code optimization (if time)
   (c) Compilers in the “real world”
      i. Compiler development for a new architecture
         A. “bootstrapping”
         B. cross compilers
      ii. Compiler development tools
         A. Scanner generators (e.g., lex)
         B. Parser generators. (e.g., yacc, bison)

2. Develop your programming abilities and organizational skills

Expectations:

1. Class participation / Communicate when things get confusing.
2. Write a compiler for a simplified language.
3. Graduate students are required to implement additional language features in their compiler project.
**Grading:** 3 exams (55%), 1 programming project, collected/reviewed in stages (45%). The compiler project **must** be submitted ready to compile and run on the SUN/Sparc platform.

**Disability Notice:**
If you have a disability that may require an accommodation for taking this course, then please contact the Learning Assistance Center (758-5929) within the first two weeks of the semester.

**Pandemic Planning Notice:**
The University has requested that faculty collect personal contact information as part of emergency planning and preparation. The information you provide is strictly confidential.