

# Computer Engineering 175

## Formal Language Theory and Compiler Construction

Winter 2010

Mondays, Wednesdays, and Fridays

1:00 pm – 2:05 pm

### Instructor

Instructor: Darren Atkinson  
E-mail: datkinson@scu.edu  
Office hours: Mondays 2:15–3:15 pm and Wednesdays 9:30-10:30 am  
Office: EC 245  
Website: <http://www.cse.scu.edu/~atkinson/teaching/wi10/175/>

### Teaching Assistant

Teaching assistant: Jingjing Ren  
E-mail: renjingj@gmail.com  
Lab hours: Wednesdays and Thursdays, 2:15–5:00 pm

### Textbooks

Required: Aho, Lam, Sethi, and Ullman, *Compilers: Principles, Techniques, and Tools*, 2007

### Grading

Midterm exam: 20%  
Final exam: 40%  
Project: 40%

### Course Outline

1. Introduction and lexical analysis
2. Syntax analysis..... Phase I due on 1/12
3. Syntax analysis..... Phase II due on 1/22
4. Semantic analysis
5. Semantic analysis ..... Phase III due on 2/2
6. Storage allocation..... Midterm exam on 2/10
7. Code generation ..... Phase IV due on 2/16
8. Code generation ..... Phase V due on 2/26
9. Formal language theory
10. Formal language theory..... Phase VI due on 3/12
11. Final exam on 3/17

## Course Objectives

Students will be able . . .

1. To understand how a compiler works, specifically the analysis of a program into atomic pieces and the subsequent synthesis into an equivalent program.
2. To gain experience in the construction of a large programming project that draws upon several previous courses.
3. To gain personal responsibility such as time management and testing for a large project.

## Learning Outcomes

Students will . . .

1. Build a compiler for a nontrivial programming language.
2. Describe the phases of compilation.
3. Specify regular expressions for matching tokens in a language.
4. Show the equivalence between regular expressions, NFAs, and DFAs.
5. Specify and disambiguate context-free grammars.
6. Specify a type system for a language including type equivalence, and use it to correctly type check expressions in a language.
7. Apply fundamentals of storage allocation strategies toward run-time management of data.
8. Generate correct assembly code for simple expressions and statements in a programming language.

## Policies

### Disability Accommodation Policy

To request academic accommodations for a disability, students must contact Disabilities Resources located on the second floor of Benson. Phone numbers are (408) 554-4111; TTY (408) 554-5445. Students must register and provide documentation of a disability to Disabilities Resources prior to receiving academic accommodations.

### Academic Integrity Policy

The University is committed to academic excellence and integrity. Students are expected to do their own work and to cite any sources they use. A student who is guilty of a dishonest act in an examination, paper, or other work required for a course, or who assists others in such an act, may, at the discretion of the instructor, receive a grade of F for the course.

In addition, a student found guilty of a dishonest act may be subject to sanctions up to and including dismissal from the University as a result of the student judicial process as described in the *Community Handbook*.

A student who violates copyright laws, including those covering the copying of software programs, or who knowingly alters official academic records from this or any other institution is subject to similar disciplinary action.