Computer Engineering 175/175L
Formal Language Theory and Compiler Construction

Winter 2019
Mondays, Wednesdays, and Fridays
10:30 am – 11:35 am and 1:00 pm – 2:05 pm

Instructor

Instructor:	Darren Atkinson
E-mail:	datkinson@scu.edu
Office hours:	Tuesdays 9:45–10:45 am and Wednesdays 9:30–10:30 am
Website:	http://www.cse.scu.edu/~atkinson/teaching/wi19/175/

Teaching Assistants

Teaching assistant:	Kush Mahajani
E-mail:	kmahajani@scu.edu
Lab hours:	Mondays 2:15–5:00 pm and Tuesdays 9:15–12:00 pm

Teaching assistant:	Kevin Velcich
E-mail:	kvelcich@scu.edu
Lab hours:	Tuesdays 2:15–5:00 pm and Tuesdays 5:15–8:00 pm

Textbook

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

Grading

Midterm exam:	20% (2/13)
Final exam:	40% (3/18 or 3/22)
Project:	40%

Course Outline

1. Introduction; lexical analysis; syntax analysis .... Phase I due on 1/13
2. Ambiguous grammars; top-down parsing of expressions
3. Parsing of statements; syntax-directed translation; semantic checking .... Phase II due on 1/27
4. Scoping and symbol table management; type checking
5. Type systems and type equivalence; storage allocation methods .... Phase III due on 2/10
6. Stack frame manipulation .... Midterm exam on 2/13
7. Intel storage allocation; code generation for simple expressions .... Phase IV due on 2/24
8. Code generation for complex expressions and statements
9. Regular languages and automata; grammars and context-free languages .... Phase V due on 3/6
10. Bottom-up parsing; decidability .... Phase VI due on 3/15
11. Final exam on 3/18 or 3/22

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.
9. Demonstrate programming proficiency in an object-oriented language (e.g., C++), including an understanding of fundamental object-oriented concepts such as encapsulation, abstraction, inheritance, and polymorphism.

Policies

Exams and Regrades

You must be registered for both the lecture and the lab. Since the project grade is more than 20% (1 unit out of 5 units) of your course grade, you simply receive the same grade for both the lecture and lab in this course. You must take the exams in the class period for which you are registered. There will be no exceptions to this policy. Any request for a regrade on any exam or assignment must be done ***in person*** through office hours or an appointment. All requests must be made within ***one week*** of the exam or assignment being returned to the class. It is not permissible to alter any work before the request for a regrade.

In-Class Recordings

The *Student Conduct Code* ***prohibits students from making a video recording, audio recording***, or streaming audio/video of private, non-public conversations and/or meetings, inclusive of the classroom setting, without the knowledge and consent of all recorded parties, except in cases of approved disability accommodations.

Disability Accommodation

If you have a documented disability for which accommodations may be required in this class, please contact Disabilities Resources, Benson 216, as soon as possible to discuss your needs and register for accommodations with the University. If you have already arranged accommodations through Disabilities Resources, please ***discuss them with me within the first two weeks of class***.

Academic Integrity

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.