

ETEC 4401: Compiler Design

Class Information

Semester Spring 2019

Time/Day T/R 9-11PM

Location ATC 306

Credit Hours 3 (2 lecture, 3 lab)

Prerequisites ETEC 3402

Description Application of finite state automata as regular expressions to programming language design and analysis of the use of context-free grammars as a formal device for language syntax. Techniques of lexical analysis and parsing (top-down and bottom-up), symbol table management, code generation, and error handling are covered.

Faculty Information

Name James Hudson

Office ATC 318

Office Hours Monday/Wednesday 2:30-4:00, Tuesday 2:00-3:00

Office Phone 740-351-3686

E-mail jhudson@shawnee.edu

Readings

There is no assigned textbook for this class. You might try reading *Crafting Interpreters* by Robert Nystrom (<http://www.craftinginterpreters.com/contents.html>).

Grading System

Labs/Homeworks/In-class work: 50%

Midterm: 25%

Final: 25%

Attendance Policy

Regular attendance is strongly encouraged. If you miss class, you are responsible for any material covered, announcements made, assignments given, or handouts distributed. Make-up exams are only given if prior arrangements have been made with the instructor. Missed homework assignments cannot be made up.

Grading Scale

The course grading scale is as follows. There is no rounding.

94-100	A	80-83.9999	B-	67-69.9999	D+
90-93.9999	A-	77-79.9999	C+	64-66.9999	D
87-89.9999	B+	74-76.9999	C	60-63.9999	D-
84-86.9999	B	70-73.9999	C-	0-59.9999	F

Academic Integrity

All assignments are to be your own work. Discussion of problems with others is encouraged, but anything you turn in should be entirely your own work. Referring to external sources (forums, web pages, etc.) for ideas is fine, but ***be sure to cite them!*** If evidence of academic misconduct is found, *all of the individuals involved will be penalized, regardless of who copied from whom.* If you disagree with the penalty received, there is an appeals process outlined in the Student Handbook.

ADA Statement

Any student who believes he or she may need an accommodation based on the impact of a documented disability should first contact a Coordinator in the Office of Accessibility Services, Hatcher Hall, 740-351-3106 to schedule a meeting to identify potential reasonable accommodation(s). Students are strongly encouraged to initiate the accommodation process in the early part of the semester or as soon as the need is recognized. After meeting with the Coordinator, students are then required to meet with their instructor's during the instructor's office hours to discuss their specific needs related to their disability. The accommodation letter will be sent to the instructor and student via e-mail prior to the semester start date. Any questions regarding the accommodations on the letter should be addressed to the Coordinator of Accessibility Services. If a student does not make a timely request for disability accommodations and/or fails to meet with the Coordinator of Accessibility Services and the instructor, a reasonable accommodation might not be able to be provided.

Objectives

Upon successful completion of this course, students should:

- Be able to design a working compiler, from lexical analysis through syntax and semantic analysis
- Demonstrate understanding of how compilers create output (either intermediate code or assembly code) from a parse tree
- Have given evidence of being able to design grammars for a variety of practical programming languages

Topic Outline

The following is a tentative list of topics for the course. This is subject to change.

- | | | |
|-----------------------|------------------|-------------------------------|
| • Regular expressions | • First | • GLR parsing |
| • Tokenization | • Follow | • Syntax Directed Translation |
| • CFGs | • LL parsing | • Interpreters |
| • Operator precedence | • LR(0) parsing | • Assembly language |
| • Left factoring | • SLR(1) parsing | • Generating efficient code |
| • Nullable | • LR(1) parsing | |