

CSCE 531: COMPILER CONSTRUCTION

Catalog Description:

531—Compiler Construction (3) (Prereq: CSCE 330 or 355, CSCE 245) Techniques for design and implementation of compilers, including lexical analysis, parsing, syntax-directed translation, and symbol table management.

Prerequisite(s) By Topic:

Programming languages, grammars, finite state automata
Logical operations

Textbook(s) and Other Required Material:

Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*, Addison-Wesley, Boston, MA, 1986 [required]

Samuel P. Harbison and Guy L. Steele, *C: A Reference Manual*, 5th edition, Prentice Hall, 2002 [recommended]

Computing Platform: Windows XP

Course Objectives: {Assessment Methods Shown in Braces}

1. Formally define the grammar and semantics of a language {homework, tests}
2. Design and implement finite state machines appropriate for use a lexical scanner {homework, programming assignments, tests}.
3. Given the definition of an appropriate context free grammar, design either a bottom-up or top-down parser for the grammar. (One type will be emphasized in a given course.) {homework, tests, programs, reports}.
4. Given the semantic definitions for an appropriate language, implement the semantic routines for a top-down or bottom up parser. (One type will be emphasized in a given course.) {homework, tests, programs, reports}.
5. Perform code generation at the tuple level {homework, tests, programs, reports}.

Topics Covered:

1. Introduction to compiler, structure, implementation, and operation (2 hours).
2. Specification, design, and implementation of a simple recursive descent compiler (8 hours).
3. Lexical analysis (3 hours).
4. Grammars and Parsing (8 hours).
5. Semantics (4 hours).
6. Symbol Tables (2 hours).
7. Run-time storage organization (4 hours).
8. Translation of language components (4 hours).
9. Code generation (3 hours).
10. Reviews and tests (4 hours)

Laboratory Projects and Other Course Work

Students write a compiler for a simple language. The project is completed in stages throughout the semester.

Difference between Undergraduate and Graduate Work:

Graduate students must complete a more complicated compiler containing additional features beyond those which undergraduate students implement.

Syllabus Flexibility: Medium. The instructor chooses the textbook, language, and projects.

Relationship of Course to Program Outcomes:

The contribution of each course objective to meeting the program outcomes is indicated with the scale:

3 = major contributor, 2 = moderate contributor, 1 = minor contributor. Blank if not related.

Course Objectives	Program Outcomes										
	1. Logic & Math	2. Computing Fundamentals	3. Apply Computing Principles	4. Work on teams	5. Communicate Effectively	6. Liberal arts & Soc. Sciences	7. Basic Science and Lab Procedures	8. Learn New Tools & Processes	9. Employed upon Graduation	10. Application Area	11. Electronics and Digital Sys Design
1. Formally define the grammar and semantics of a language		3	1						1		
2. Design and implement finite state machines appropriate for use a lexical scanner		1	3						1		
3. Design a bottom-up or top-down parser for a given grammar		1	3					1	1		
4. Implement the semantic routines for a top-down or bottom up parser.		1	3					1	1		
5. Perform code generation at the tuple level		2	3					1	1		

Estimated Computing Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Algorithms		1	Data Structures		
Software Design		1	Programming Languages		1
Computer Architecture					

Estimated Information Systems Category Content (Semester hours):

Computer information systems majors do not take this course.

Oral and Written Communication: None

Social and Ethical Issues: None

Theoretical Content:

Formal grammars and languages

Analysis and Design:

Program organization and structure, symbol table design, design of a parser.

Class/Laboratory Schedule:

Lecture: 3 periods of 50 minutes or 2 periods of 75 minutes per week

Course Coordinator: Stephen Fenner

Modification and Approval History:

Initial description April 1999

Revised June 2001

Revised June 2005 by Caroline Eastman to update textbook and revise format