Course Outcomes	CSC 4351

CSC 4351: Compiler Construction

Credit Hours: 3 hours

Prerequisites:

CSC 4101 or equivalent

Prerequisites by Topics:

Basic syntax and semantics of programming languages, object-oriented programming.

Catalog Course Description:

Program language structures, translation, loading, execution, and storage allocation; compilation of simple expressions and statements; organization of compiler including compile-time and run-time symbol tables, lexical scan, syntax scan, object code generation, error diagnostics, object code optimization techniques, and overall design; use of compiler writing languages and bootstrapping.

Course Outcomes

1. Master using lexical analyzer and parser generator tools.
2. Master building symbol tables and generating intermediate code.
3. Master generating assembly code for a RISC machine.
4. Master programming in Java.
5. Be familiar with compiler architecture.
6. Be familiar with register allocation.
7. Be exposed to compiler optimization.

Texts and Other Course Materials

- Andrew W. Appel, *Modern Compiler Implementation in Java*. Cambridge University Press, 1998 or 2002 (required).
- Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam, *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 2006 (optional).
- Thomas W. Parsons, *Introduction to Compiler Construction*. Computer Science Press, 1992 (optional).
- http://java.sun.com/docs/books/tutorial/
- http://java.sun.com/docs/books/jls/
- http://java.sun.com/j2se/1.5/docs/api/index.html

Major Topics

- Compiler architecture.
- Lexical analysis and scanner generation tools.
- Parsing (top-down and bottom-up parsers) and parser generation tools.
- Symbol tables and semantic analysis.
- Activation record representations.
- Intermediate code generation and canonicalization.
- Instruction selection.
- Liveness analysis and register allocation.

Assignments/Projects/Laboratory Projects/Homework

- Implement a lexical analyzer.
- Implement a parser.
- Implement a type checker.
- Implement activation record generation.
- Implement translation to intermediate code.
- Implement assembly instruction selection.

Curriculum Category Content (estimated in semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	10	8	Data Structures	5	5
Software Design	2	2	Prog. Languages	2	
Computer Arch.	3				

Relationship to Criterion 3 Outcomes

A	B	C	D	E	F	G	H	I	J	K
*	*	*		*		*		*		*

Math and Fundamentals:

Regular expressions, grammars.

Data Structures:

Tokens, parse trees, symbol tables, type representation, intermediate code, frames, abstract assembly, basic blocks (10 hrs).

Algorithms and Software:

Analysis of lexical analysis and parsing algorithms (1 hr). Design and implementation of lexical analysis, parsing, semantic analysis, intermediate code generation, canonicalization, instruction selection, register allocation, data flow analysis (21 hrs).

Computer Organization and Architecture:

Calling sequence, assembly (3 hrs).

Concepts of Programming Languages:

Semantics of Java and Tiger.

Social and Ethical Issues:

None.

Oral Communication (presentations)

None.

Written Communication:

4 homeworks, 6 programming assignments.

Course Coordinator: Dr. Gerald Baumgartner

Last Modified: June 10, 2007.