

CIS 650 Compiler Design

Course Description: (Prerequisites: CIS335 and CIS345)

Practical overview of the principles involved in the design and construction of compilers. Understand formal language theory, parsing methods, and gain experience in constructing certain compiler components using lex/flex and yacc/bison.

Course Outcomes: After completing this course, students should be able to:

- Specify regular expressions for matching tokens in a language
- Write LL(1), LR(1), and LALR(1) grammars for new languages
- Code parser using construction tools or directly
- Understand attribute grammars, semantic actions, and attribute evaluation
- Understand data-flow analysis and optimization techniques

Textbook: Compilers Principles, Techniques and Tools, 2nd ed.,
by Aho, Lam, Sethi, Ullman,
Addison Wesley

Reference: lex and yacc,
by J. Levine, T. Mason, D. Brown,
O'Reilly & Associates Inc.

Instructor: Janche Sang
Email: sang@cis.csuohio.edu
Phone: 687-4780

Grading Policy:	Projects, Homeworks, Quizzes & Class Participation	35%
	Midterm	30%
	Final	35%

The course grade is based on a student's overall performance through the entire semester. The total points will be curved. No makeup exams will be given unless notified and agreed to in advance. No makeup quizzes.

Other Information:

- Late projects cannot be accepted for any reason. All assignments are due at the beginning of class on the date they are due.
- Students are encouraged to discuss homeworks/projects with classmates. However, each student must do his/her own work. Evidence of copying will result in a zero grade for all students involved. Also, students are responsible for protecting their own programs and homeworks.

Academic misconduct and cheating will not be tolerated. Violations will be subject to disciplinary action as specified in the CSU Student Conduct Code.

- Programming assignments are to be done on one (or more) of the department UNIX work-stations such as merlin (LINUX), mermaid (Sun Solaris), etc. The department server, grail, may not be used! Note that these UNIX machines have been shut off from outside access by IS&T with the exception of grail. Therefore, you have to use SSH secure shell to login grail or spirit and then access the other workstations.

Tentative Course Outline:

1. Introduction (Chapter 1)
2. A Simple Translator (Chapter 2)
3. Lexical Analysis (Chapter 3)
 - finite automata and lexical scanner lex/flex
 - lex
4. Syntax Analysis (Chapter 4)
 - top-down recursive-descent parsing
 - bottom-up parsing
 - LR parsers
 - yacc/bison
5. Syntax-Directed Translation (Chapter 5)
 - synthesized attributes and inherited attributes
 - L-attributed definitions
6. Intermediate Code generation (Chapter 6)
 - three-address code
 - type declaration
 - type checking
7. Run-Time Environment (Chapter 7)
 - stacks and activation records
8. Code Generation and Optimization (Chapter 8 and Chapter 9)
 - dag representation of basic blocks
 - register allocation and assignment
 - data flow analysis