

Brief Syllabus of COSC 4307-01, Summer I 2005

Compiler Construction

Mon ~ Fri, 8:00 ~ 9:20 PM, Maes 111

Instructor: Dr. Chung-Chih Li
Office: Maes 69, Tel: (409) 880-8748
E-mail: licc@hal.lamar.edu Class URL: TBA
Office Hours: MWF 11:00 ~ 11:50 AM

Description of the course and Topics:

By now, you should have learned a lot of data structures, programming techniques, and theory. For examples, you should have been familiar with the structures of linked lists, trees, and graphs from the data structures courses; the techniques of recursion, backtracking, and dynamic programming from the algorithm course; the concepts of finite state machines, automata, and formal languages from the theory course. But, you may also feel these fancy techniques seem so remote for real applications. In this course, we will put them together and use them to construct a very practical computer tool – *compiler* – a bridge between human thinking (the logic of solving problems in terms of programming languages) and machine codes (the logic of instructing machine operations). We will explore the major components of classical compilers: Scanners, Parsers, and Code Generators – they serve as a step by step translation from a program to machine codes. We will study Regular Expressions, Context-Free Languages, and Context-Sensitive Languages that play indispensable roles in understanding the underlying theory and limitation of a compiler. Also, we will examine two extra components of most modern compilers: Intermediate Code Generators and Optimizers; as such, Intermediate Representations, Data-Flow Analysis, and some scheduling problems will also be introduced in some detail.

Prerequisites:

CS1, CS2, CS3 (COSC 2336, Data Structures), and with certain maturity of programming skill.

Textbooks: (required)

Modern Compiler Implementation in JAVA, by Andrew W. Appel, Cambridge University Press, 2nd edition, 2002.

References: (optional)

The following texts only help you when certain topics presented in our textbook are not clear to you; **they are not surrogate of the textbook.**

1. Engineering a Compiler, by Keith D. Cooper and Linda Torczon, Morgan Kaufmann Publishing, 2004.
2. Compilers: Principles, Techniques, and Tools, by Alfred Aho, Ravi Sethi, and Jeffrey Ullman, Addison Wesley, 1988.
3. Crafting a Compiler with C++, by Charles N. Fischer, et al, Benjamin/Cummings, 1996

Tests: (300 points)

Two midterms and one final, 100 point for each.

Programming Assignments: (300 points)

About 3 programming assignments will be given. The weight of each program depends on its difficulty.

Attendance and Pop quizzes: (100 points)

Attendance and pop quizzes will be taken and given impulsively.

Grading Policy:

Your points will be accumulated and the grade is based on the following scheme.

Points	Grade	
540 ~	A	Excellent
420 ~ 539	B	Good
300 ~ 419	C	Satisfactory
200 ~ 299	D	Passing
0 ~ 199	F	Failure

I do not curve!!