

CS 471 – Intro to Compiler Construction

Fall 2007

Time: Monday & Wednesday @ 11:30 AM
Location: MEC 216
Instructor: Kim Hazelwood (hazelwood AT virginia.edu)



This course is intended to provide a general introduction to the field of compilers, which translate programs written in high-level languages to other low-level codes. It will cover theories and mechanisms of compilation tools. Students will learn the core ideas behind compilation and how to use software tools such as lex/flex, yacc/bison to build a compiler for a non-trivial programming language.

Course Topics

- Structure of Compilers
- Lexical Analysis
- Parsers
- Attribute Grammars, Actions, and Parse Trees
- Scoping in Language Definition
- Symbol Tables
- Type Checking
- Run-time Data Structures
- Intermediate Code
- Code Generation
- Dataflow Analysis
- Liveness Analysis and Register Allocation

Prerequisites

C+ or better in CS 333 and CS 340 is required. A strong programming background will be important to your success in this course.

Grading

- 20% - In-Class Exam 1
- 20% - In-Class Exam 2
- 60% - Compiler Implementation Assignments

Assignments

This semester we will be building a compiler for the Tiger programming language. The project will be broken up into roughly 10 subprojects. The primary purpose of the project is to re-enforce the concepts discussed in lecture. The project also provides you with hands-on experience constructing a compiler using software engineering concepts.

Exams

We will have non-cumulative in-class exams on October 3 and November 28. In lieu of a final exam, there will be a robustness competition between all of the compilers developed in class.

Required Text

Andrew Appel. **Modern Compiler Implementation in C**, Cambridge Press, 2004. ISBN **0-521-60765-5**

Late Policy

Late work will result in the following point deduction: $2^{\text{(number of days late)}}$. Furthermore, you will have fewer days to complete the next portion of the project, as each assignment builds on the previous assignment.

Cooperation

Please feel free to talk with other class members about the algorithms in the programming assignments, but you should **never** even see someone else's code. Any evidence of a violation of this policy will result in an F for the course, and a case with the honor board. It is in your best interest to attend the lectures and to keep pace with the project assignments.