

CS 488 Compilers

Syllabus

Spring 2007

Lectures: MW 2:30-4:10PM, E&T A331

Instructor: Valentino Crespi
vcrespi@calstatela.edu
(323) 343-4596
E&T A318

Office Hours: TBA

Course Description: Compiler construction; lexical analysis, including regular languages and finite-state acceptors; syntactic analysis, including parsing techniques and grammars; code generation and optimization.

Course Goals: At the end of the course, students get an understanding of:

- 1 The organization of a compiler
- 2 The concepts of scanning, parsing, and translation
- 3 Compiler writing tools
- 4 Modular development of a significant programming system

These course goals contribute to the success of **Student Learning Outcomes 1.a, 1.c, 2, 3, 5, and 6.**

Prerequisites: CS 312 Data Structures and Algorithms
CS 332 Programming Language Paradigms
CS 386 Introduction to Automata Theory

Textbook(s): *Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman : Compilers - Principles, Techniques, and Tools*

Reference(s): 1 Kenneth C. Loudon, Compiler Construction, Principles and Practice, PWS Publishing Company, 1997

- Topics:**
- 1 Introduction to compiler design.
 - 2 Symbol tables.
 - 3 Lexical analysis.
 - 4 Syntactic analysis – top down and bottom up parsing scheme
 - 5 The use of compiler-writing tools: automated parsers and lexical analyzers.
 - 6 Error recovery.
 - 7 Semantic analysis.
 - 8 Translation of source to an intermediate language.
 - 9 Translation of intermediate language to object code.
 - 10 Optimization of object code.

Lab Project: The students will complete an implementation of a compiler for a mini language and the are required to demonstrate the compiler on a few algorithms at the end of the quarter. The project is divided into five phases:

- 1 Phase 1 : (1 week) Report on the chosen grammar and implementation issues including the language and the system.
- 2 Phase 2 : (2 weeks) Implementation of a Lexical Analyzer
- 3 Phase 3 : (3 weeks) Implementation of a Syntax Analyzer
- 4 Phase 4 : (2 weeks) Implementation of a Semantic Analyzer and an Intermediate Code Generator
- 5 Phase 5 : (1 week) Implementation of an interpreter and testing the compiler implementation on a few algorithms.

Grading Policy: Homework and Projects 50%, Midterm 30%, Final 20%

A-,A	90 - 100
B-,B,B+	80 - 89
C-,C,C+	65 - 79
D,F	<65

Academic Integrity: Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and be reported to the Academic Senate.

