

COURSE DESCRIPTION

Department and Course Number	CS 488	Course Coordinator	Valentino Crespi
Course Title	Compilers	Total Credits	4

Current Catalog Description:

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

Textbook:

Aho, Alfred V. and Sethi, Ravi and Ullman, Jeffrey D. *Compilers - Principles, Techniques, and Tools*, Addison Wesley,

References:

Louden, Kenneth C. *Compiler Construction, Principles and Practice*, PWS Publishing Company, 1997.

Course Goals:

At the end of the course, students have a good understanding of

- The organization of a compiler
- The concepts of scanning, parsing, and translation
- Compiler writing tools
- 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 by Topic:

- Data Structures
- Algorithms
- High level programming languages
- Formal languages
- Automata theory

Major Topics Covered in the Course:

- Introduction to compiler design.
- Symbol tables.
- Lexical analysis.
- Syntactic analysis – top down and bottom up parsing schemes.

- The use of compiler-writing tools: automated parsers and lexical analyzers.
- Error recovery.
- Semantic analysis.
- Translation of source to an intermediate language.
- Translation of intermediate language to object code.
- Optimization of object code.

Laboratory Project (specify number of weeks on each):

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

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

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms		0.5	Data Structures		0.5
Software Design		2.0	Prog. Languages		1.0
Comp. Arch.					

Oral and Written Communications:

The students are required to submit the source code and software documentation of the programming project. The students present the project orally.

Social and Ethical Issues:

No significant component.

Theoretical Content:

- Regular expressions and finite automata and automatic generation of lexical analyzers (1 week)
- Formal techniques for parsing context-free languages (2 weeks)

Problem Analysis:

In order to write a compiler, students must analyze how the techniques discussed in class relate to the particular language to be processed

Solution Design:

Students implement a compiler for a simple programming language using object-oriented techniques and modular design.