

CSC 490 Compiler Construction (formerly CSC 390)

4 cr.

Instructor: TBA
email: TBA@salemstate.edu

Office: location
Office Hours: days and times

Phone: (978) 542-extension

Section	Time	Room	Final Exam
nn	days and times	location	date and time
Lnn	days and times	location	

Catalog description:

The fundamental problems in the design and implementation of programming language processors are studied. Language syntax and semantics, parsing, implementation techniques such as recursive descent and backtracking, code generation, optimization, and error diagnostics are covered. Concepts are illustrated through extensive programming assignments. Three lecture hours per week and three hours of scheduled laboratory per week, plus extensive laboratory work outside of class.

Prerequisite: CSC 260.

Goals:

This course is intended to give students an in-depth introduction to programming language processors as an important subdiscipline of computer science and as an example of a problem solution that has achieved orders of magnitude improvement through the rigorous and methodical study of the problem.

- CG01: understand the environment and basic features of a compiler;
- CG02: understand the use of grammars in specifying a language;
- CG03: be familiar with the phases of compilation and how they are implemented;
- CG04: understand how object oriented programming techniques are applied to a compiler.

Objectives:

Upon successful completion of the course, a student will have demonstrated the ability to:

- CO01: analyze and describe the compiler environment;
- CO02: specify, analyze and transform grammars describing programming languages;
- CO03: understand concepts of the various compiler phases, and analyzed and modified implementations of compiler phases;
- CO04: program a simple compiler or fragment and extend a larger complete compiler.

Student Outcome (SO) vs. Course Objectives matrix

Student Outcome	CO01	CO02	CO03	CO04
SO-1	✓	✓	✓	
SO-2		✓	✓	✓
SO-3				
SO-4				
SO-5				
SO-6	✓	✓	✓	✓

Notes:

SO-1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

SO-2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

SO-3: Communicate effectively in a variety of professional contexts.

SO-4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

SO-5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. Apply computer science theory and software development fundamentals to produce computing-based solutions.

SO-6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics:

- Introduction: **PL6(0, 2, 0),**
 - compilers, interpreters, assemblers
 - single-pass vs. multiple-pass compilers
- Overview: lexical analysis, parsing, code generation **PL7(0, 0, 1), PL8(0, 0, 1), PL9(0, 0, 1)**
- Representing computer languages: **PL7(0, 0, 2), PL8(0, 0, 2)**
 - EBNF, grammars, parse trees, syntax diagrams (or syntax trees)
 - Example: a simple compiler (e.g., a recursive-descent expression compiler)
- Scanning and tokenizing **PL6(0, 6, 0)**
- Grammars (especially context-free grammars) **AL4(1, 1, 0), PL12(0, 0, 1)**
- Parsing (top-down vs. bottom-up) **PL6(0, 2, 0)**
- LL parsers **PL6(0, 1, 0)**
- Symbol tables **PL5(2, 1, 0)**
- Semantic processing **PL8(0, 0, 2)**
- Storage management **PL6(0, 1, 0), PL9(0, 0, 2)**
- Handling types and variables **PL4(1, 3, 0)**
- Processing expressions **AL4(1, 1, 0), AL6(0, 0, 1)**
- Translating control structures **PL6(0, 1, 0)**
- Translating records **PL6(0, 1, 0)**
- Translating arrays **PL6(0, 1, 0)**
- Optimization strategies **PL9(0, 0, 1)**

Assignments:

A number of written assignments will involve practice with EBNF, parse trees, syntax diagrams, etc. Programming assignments will involve the design and implementation of portions of a compiler for a subset of a specific programming language such as Java or C. Consult the instructor for the language(s) to be studied and used in a given semester.

Examinations:

Normally two examinations are given in class at the one-third and two-thirds points of the course. There is also a comprehensive written final examination.

Grades:

The course grade will be determined using the following approximate weights: lab assignments - 10%, programs - 30%, exams - 50%, homework - 10%.

Course Objective / Assessment Mechanism matrix

	Lab Assignments	Programming Assignments	Homework	Exams
CO01			✓	✓
CO02			✓	✓
CO03	✓		✓	✓
CO04		✓		

Bibliography:

Aho, Sethi & Ullman. **Compilers, Principles, Techniques and Tools. Second Edition.** Addison-Wesley, 2006.
Appel, Andrew; Ginsburg, Maia. **Modern Compiler Implementation in C.** Cambridge University Press, 2004.
Fischer, Charles; LeBlanc, Richard. **Crafting a Compiler in C.** Benjamin-Cummings, 1991.
Fischer, Charles; Cytron, Ronald; LeBlanc, Richard. **Crafting a Compiler.** Addison-Wesley, 2010.
Kaplan, Randy. **Constructing Language Processors for Little Languages.** Wiley, 1994.
Mak, Ronald. **Writing Compilers and Interpreters: a Software Engineering Approach.** Wiley, 2010.
Metsker, Steven J. **Building Parsers with Java.** Addison-Wesley, 2001.
Watt & Brown. **Programming Language Processors in Java.** Prentice Hall, 2000.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put, **do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.
