

COMPSCI 450/550: Programming Language Translation

Instructor

Instructor: Jim Buffenbarger
Electronic mail: buff@cs.BoiseState.edu
Office: MEC-302C 426-3567

Meetings

Lectures: MoWe 4:05–5:45 MEC-309
Office hours: MoWe 3:00–4:00 MEC-302C
by appointment MEC-302C

Catalog Description

Theory and practice of formal language translation, experience with compiler construction tools under Unix. Students work on significant projects.

Objectives

Students are introduced to basic concepts of programming-language translation, including:

- lexical analysis
- syntax analysis
- syntax-directed translation
- type checking
- code optimization
- code generation
- run-time environments
- practical tools, their formal models, and underlying algorithms

Prerequisites

COMPSCI 125 Introduction to Computer Science I
COMPSCI 225 Introduction to Computer Science II
COMPSCI 253 Object-Oriented Development in C
COMPSCI 354 Programming Languages

In addition, familiarity with Unix, Pascal, C, C++, and Java is assumed.

Textbook and Other Resources

The textbook is:

Compilers: Principles, Techniques, and Tools, by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. Second Edition. Addison-Wesley, 2007, ISBN-10: 0321547985, ISBN-13: 9780321547989.

Other resources can be found at:

<http://cs.boisestate.edu/~buff/links/sw.html>

Activities

Grades are based on student performance of several activities. They are listed below, along with their relative weights.

Homework	50%
Exam	25%
Final	25%

Homework

Several homework sets are assigned during the semester. Most are relatively large programming assignments.

Exams

An exam and a final are administered during the semester. These are in-class, open-note, and open-textbook (but no other books) tests.

Documentation Standards

Good documentation and programming style is very important. Your programs must demonstrate these qualities for full credit. Good documentation and programming style includes:

- heading comments giving: author, date, class, and description
- function/procedure comments giving description of: purpose, parameters, and return value
- other comments where clarification of source code is needed
- proper and consistent indentation
- proper structure and modularity

When you submit a program, include: the source code, sample input data, and its corresponding results.

Grading

Homework is to be submitted at the beginning of lecture on the day it is due. Late work is typically not accepted. Try to let me know, ahead of time, if you will miss an examination.

Scores are posted near my office, as they become available. You are encouraged to check your scores to ensure they are recorded properly.

Academic Honesty

The following quotation is from the BSU Undergraduate Catalog. You should read that section.

The university's goal is to foster an intellectual atmosphere that produces educated, literate people. Because cheating and plagiarism are at odds with that goal, they shall not be tolerated in any form. Therefore, all work submitted by a student must represent that student's own ideas and effort; when the work does not, the student has engaged in academic dishonesty.

There is related material in the BSU Student Handbook.

The course instructor is responsible for handling each case of academic dishonesty in the classroom except where a major or repeated offense is involved. In a proven case of cheating a student will be dismissed from the class and a failing grade issued.

There are many forms of academic dishonesty. Some relevant examples include:

- Submitting programs, or parts of programs, written by someone else.
- Viewing exam answers, homework answers, or programs written by someone else. This includes material from other courses and previous semesters.
- Distributing exam answers, homework answers, or programs to someone else, even after it has been graded.

The BSU Undergraduate Catalog contains more examples. If you are unsure about a particular case, ask your instructor,

On homework, a student must work independently. Ideas and general principles can be discussed with other students, but work must be original.

Keep your files to yourself. See the Unix commands `chmod go-rwx` and `ls -l`.

On exams, of course, each student must work entirely independently.

Computer Accounts

Each student receives an account on the department's network of computers, which run the Linux operating system. If you are unfamiliar with the department's computers, you are urged to attend office hours during the first week or two of classes. I'll try to get you started with these powerful tools.

You are responsible for understanding and complying with the departmental computing policy.

Schedule

<i>Week</i>	<i>Date</i>	<i>Topic</i>	<i>Assigned</i>	<i>Due</i>	<i>Reading</i>
1	Jan 21 Mon	MLK Day			
	Jan 23 Wed	Introduction			1
2	Jan 28 Mon		HW#1		
	Jan 30 Wed	A Simple Translator			2
3	Feb 04 Mon				
	Feb 06 Wed				
4	Feb 11 Mon		HW#2	HW#1	
	Feb 13 Wed				
5	Feb 18 Mon	President's Day			
	Feb 20 Wed				
6	Feb 25 Mon	Lexical Analysis			3
	Feb 27 Wed				
7	Mar 04 Mon				
	Mar 06 Wed				
8	Mar 11 Mon		HW#3	HW#2	
	Mar 13 Wed	Syntax Analysis			4
9	Mar 18 Mon				
	Mar 20 Wed		HW#4	HW#3	
10	Mar 25 Mon	Spring Break			
	Mar 27 Wed	Spring Break			
11	Apr 01 Mon				
	Apr 03 Wed				
12	Apr 08 Mon				
	Apr 10 Wed	Exam			
13	Apr 15 Mon		HW#5	HW#4	
	Apr 17 Wed				
14	Apr 22 Mon	Syntax-Directed Translation			5
	Apr 24 Wed				
15	Apr 29 Mon	Intermediate-Code Generation			6
	May 01 Wed				
16	May 06 Mon	Run-Time Environments		HW#5	7
	May 08 Wed				
17	May 15 Wed	Final: 5:00–7:00			