Syllabus
CMSC 445: Compiler design

Spring 2012

Time:	MWF 2:00
Room:	Ruffner 352
Website:	`http://cs.longwood.edu/courses/cmsc445`

A course covering the basic theory and techniques of compiler and code translation systems. Topics include lexical analysis, parsing, and code generation, and the various techniques used when handling differing source language classes. A programming intensive course. Also covered are the techniques of top-down and bottom-up parsing.

Professor:	Don Blaheta
Office:	Ruffner 337
Phone:	x2191
Email:	`blahetadp@blahedo.org` [1]
Office hours:	Mondays 11–noon; Wednesdays 4–5:30; Thursdays 10–11am and by appointment

Overview

Unless you go to work for a compiler company or do research in debugging or optimisation, it's relatively unlikely that you will be called upon to implement a traditional compiler. However, the main compiler components—lexing, parsing, and code generation—crop up separately in a much wider variety of applications; and a better understanding of the compilation process will make you a better programmer overall (and much better at deciphering some of the more cryptic error messages that compilers give from time to time).

Plus, there's a special thrill in running a compiler that you wrote yourself, and then running the executable it produces!

[1] Or `blahetadp@longwood.edu` if you'd rather, but I prefer the off-campus one and check it more frequently.

Objectives

At the end of this course, the successful student will be able to:

1. identify and explain the steps of the compilation process;
2. implement a compiler;
3. use automated compiler construction tools; and
4. understand and explain different strategies for lexing, parsing, and code generation.

Textbook and resources

The primary, required textbook for this course is Aho, Lam, Sethi, and Ullman, *Compilers: principles, techniques, & tools*, 2e. (ISBN 978-0-321-48681-3.)

In addition, I recommend (but do not require) the following two books as well-written and useful references for your use as we get into the project:

- Prinz and Kirch-Prinz, *C pocket reference.*
- Levine, Mason, and Brown, *flex & bison* OR its earlier edition (same authors), *lex & yacc.*

All our implementations will be done on the departmental Linux systems. If you don't have an account or have forgotten your password, see me soon to get this resolved.

This term I am trying out an online message board system called Piazza. It is specifically geared to enable out-of-class Q&A about course material, and it appears considerably less clunky than some others that I've seen. You'll get an invite shortly; I encourage you to poke and prod and tell me what you think of it, because if it works well I'd like to use it for other classes too.

Content

Calendar

Week	M	W	F
	January		
1		**18** (§§1.2, 1.3) Compile process History	**20** §1.6 Programming language basics
2	**23** §§2.0–2.2 Syntax	* **25** §§2.3–2.5 Parsing Syntax-driven translation	**27** §2.8 Representation Code generation
		February	
3	**30** §§2.6–2.7 Lexical analysis Symbol tables	**1** §§3.0–3.2 Lexical analysis Buffering	**3** §§3.3–3.4, 3.6 Regular expressions State machines
4	**6** §§3.7, 3.5 Constructing FAs `lex`	**8** §§4.0–4.2 Syntax CFGs, BNF	**10** §§4.3–4.4.2 Ambiguity Recursive descent
5	**13** §§4.4.3–4.4.5 Rec. desc. (cont'd) LL(1)	**15** §4.5 Bottom-up parsing Shift-reduce conflicts	**17** §4.6 LR parsing
6	**20** §§4.7.0–4.7.3 LR(1)	**22** §§4.7.4–4.7.6 LALR	**24** §§4.8–4.9 `yacc`
			March
7	**27** — Review/makeup	**29** — Review/makeup **Exam 1 out**	**2** [Take-home exam **no class**]

* **24 January**: Deadline to add/drop classes (5pm)

Week	M	W	F
	March		
8	**5** §§5.0–5.2 Syntax-directed translation **Exam 1 due**	* **7** §§5.3–5.4 SDT cont’d	**9** §5.5 SDT cont’d
		SPRING BREAK	
9	**19** §§6.1–6.2, 6.4 Expressions	**21** §§6.3, 6.5 Declarations Types	**23** §§6.6 Control
10	**26** §§6.6–6.8 Control (cont’d) Backpatching	**28** — x86 assembly	**30** — x86 assembly
	April		
11	**2** §§8.0–8.2 Code generation	**4** §§8.3–8.4 Addresses Basic blocks	**6** §8.6 Generating assembly
12	**9** §§7.0–7.2 Activation stack Stack allocation	**11** — x86 stack conventions	**13** §7.3 Nonlocal data
13	**16** §7.4 Heap allocation	**18** TBA**	**20** TBA**
14	**23** TBA**	**25** TBA**	**27** TBA** **Exam 2 out**
15	**Exam 2 due**: Friday, 4 May, 10:30am		

* **6 March**: deadline to withdraw from a class (5pm)

** **TBA** topics may include optimisation, garbage collection, or parallelism.

Grading scale

I tend to grade hard on individual assignments, but compensate for this in the final grades. The grading scale will thus be approximately as follows:

A−	[85, 90)	A	[90, 95)	[2]	
B−	[70, 85)	B	[75, 80)	B+	[80, 85)
C−	[55, 60)	C	[60, 65)	C+	[65, 70)
D−	[40, 45)	D	[45, 50)	D+	[50, 55)

While there will be no "curve" in the statistical sense, I may slightly adjust the scale at the end of the term if it turns out some of the assignments were too difficult.

Graded assignments

I figure that I have about 8–10 hours of your time every week, including class time as well as reading assignments, homework, and the project. If you find you're spending more time than this, please do come discuss it with me, and we'll see what we can work out. The work you do for this course will be evaluated as follows:

Preparation and participation. You'll need to read (and think about!) a portion of the textbook before nearly every class. You need not perfectly understand the reading—that's what class is for—but you should be able to identify confusing parts and be ready to ask questions about them. Reading quizzes will fall into this category. Note that attendance is a prerequisite: if you aren't there, you may or may not be prepared but you definitely aren't participating. Preparation and participation are collectively worth 5% of the grade.

Project. The bulk of the programming you will do for this course will be for the task of building a compiler. It will be broken up into several discrete chunks to aid in pacing (and grading), but you'll generally be expected to use the result of one phase as the basis for the next. Taken together these assignments will make up 30% of the grade.

Homework. For more theoretical work, there will be written homework assignments and the occasional short program not related to the project. These will make up 25% of the grade.

Exams. There will be two take-home exams, one at the start of March and one during the finals period. The final will not be explicitly cumulative, though of course the material from the second half of the course builds on the earlier stuff. Each exam is worth 20% of the grade.

[2] Alas, no A+. Pity.

Policies

Support

Don't hesitate to talk to me if you're having trouble with any part of the class. I can help—it's my job! I'm in my office a lot (not just during posted office hours). Feel free to come in and ask questions (or just to talk). If you can't catch me in my office, email is a good bet (although see above about Piazza—try posting your question there!).

Accommodations

If you have any special need that I can accommodate, I'm happy to do so; come speak to me early in the term so we can set things up. If you have a documented disability, you should also contact Longwood's Office of Disability Resources (Graham Hall, x2391) to discuss some of the support the college can offer you. All such conversations are confidential.

Collaboration policy and honor code

Exams are non-collaborative: no discussing them with anyone else (just me). Homeworks and papers are what I call "lightly collaborative"; the short version is that you can talk to other students about it, but you must write it yourself. For a fuller description of what this means, and some examples of acceptable and unacceptable behaviour, please go to
`http://torvalds.cs.longwood.edu/~dblaheta/collab.html` . As in any other class, material substantially derived from another source (e.g. quoted from the book, strongly inspired by a classmate) should be cited, if only in a short comment.

Violations of this policy (on any assignment) will normally receive a *minimum* penalty of a lowered *course* grade, ranging up to an F in the course. Cases will also be turned in to the Honor Board. You should consider all work in this class to be pledged work.

Attendance and late policy

Attendance is required, and assignments must be turned in on time. That said, if you have a good reason to miss class or hand something in late, I tend to be fairly liberal with extensions if you ask in advance. (Good reasons do include assignments due for other classes.) (And medical and family emergencies are exempted from the "in advance" part, of course. But get a note from a dean.)

Frequent absence will result in a lowered participation grade; habitual absence may in extreme cases result in a failing grade for the class. *Unexcused* late assignments will normally be given a zero.

Inclement weather policy

I don't plan to cancel class for weather unless the entire college shuts down. If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement; and if you do miss class for this reason, contact me as soon as possible to make up missed work.

Early bird policy

Nobody's perfect, and on occasion an assignment gets written a little unclearly (or, once in a while, with an actual error in it). If you catch one and bring it to my attention early, so that I can issue a clarification or correction to the rest of the class, there'll be some extra credit in it for you.