

CS143 Compilers Syllabus

Instructor: David Underhill
E-mail: dgu@cs.stanford.edu
Cell Phone: 304-541-2399
Office Hours: Gates B26A on Tuesdays, Fridays 9:30 – 11:30 a.m.

TA: Ian Goodfellow (ia3n@stanford.edu)
Office Hours: Gates B24A on Wednesdays, Thursdays 3:00 – 5:00 p.m.

Lecture: Online only; three will be released each Tuesday (starting 24-Jun-2008)
Video available at: scpd.stanford.edu

Section: Tuesday 4:15 – 5:05 p.m. (starting 01-Jul-2008)
Skilling 191 (Broadcast Live on SCPD Channel TBD)

The Class: CS143 is a first course in compilers, designed to introduce the student to the principles and practices of programming language implementation. We cover lexical analysis, parsing theory (LL, LR, and LALR parsing), semantic analysis, runtime environments, code generation, and optimization. The coursework will consist of regular programming projects, a few pencil-and-paper problem sets, a midterm and a final exam. The four programming projects comprise a series of milestones that will ultimately lead to an operational compiler for a miniature object-oriented programming language.

Having traced through and implemented the various phases of compilation, you will gain a clear understanding of how a compiler works. You will grow to appreciate the difficulties and trade-offs involved in implementing a compiler or an interpreter for any programming language. You will gain experience with commonly used tools in compiler development and learn standard techniques that can be applied to a variety of parsing problems. Exposure to programming language implementation will strengthen your development and debugging skills and generally aid your understanding of language and programming issues. CS143 is one-third theory, two-thirds implementation, and the algorithms we discuss in lecture map almost exactly to the way actual tools work.

The Student: The prerequisites are CS103B/X and CS107. You should be familiar with the CS103B content: regular expressions, graphs, sets, context free

grammars, and formal languages. If you have never heard of these things before you can still take the course, but you must learn that material on your own. You should have well-developed programming and debugging skills, understand simple code generation and function calling conventions, and have some exposure to different language paradigms. The projects will require you to manage unfamiliar tools, complex algorithms, and advanced programming techniques. It should go smoothly if you know how to code as well as a CS107 graduate.

Sections: Starting on Tuesday 01-Jul-2008, we will lead one discussion section every Tuesday (broadcast live on SCPD) in Skilling 191. It will also be available online via scpd.stanford.edu. Section attendance is not required, but you can only benefit by participating. All section materials will be posted as handouts on the course web site.

Units: CS143 is offered for three or four units. Undergraduates must take the course for four units of credit. Graduate students may enroll for three units. The course requirements are the same for everyone.

Handouts: All handouts will also be posted on the course web site prior to the beginning of the discussion section.

Textbook: There will be some readings assigned from Compilers: Principles, Techniques, and Tools by Aho, Sethi, and Ullman (a.k.a. the "red dragon" book). This book is the classic compiler text that provides a very thorough treatment of the material we will cover this quarter and more.

I also highly recommend (but do not require) lex & yacc by John Levine, Tony Mason, Doug Brown (second edition). It is both a good tutorial and reference for two of the new tools we will be using this quarter.

Web: This site is the place to grab handouts, read late-breaking announcements, scan the syllabus, and get general course information. We will also maintain a FAQ page with common question/answers, clarifications, etc. about the current assignment. You should also check the web page periodically for course announcements.

Questions and E-mail: Unless you need to address me specifically, you should post questions on the course newsgroup (su.class.cs143). For private matters, you may send e-mail to both Ian and I by e-mailing compilers@cs.stanford.edu.

Mailing Lists: We will send important announcements to the CS143 mailing list. All students enrolled in CS143 via Axess are automatically subscribed to the `cs143-aut0708-students@lists.stanford.edu` mailing list. You should inspect your `stanfordyou.stanford.edu` privacy settings to ensure that **some** email address is marked as public. If not, you are excluded from the mailing list, and you will miss out on all the gossip.

Coursework: The assigned coursework consists of a sequence of programming projects with two written problem sets mixed in. Over the course of the quarter, you will build a working compiler for a simple object-oriented language. History shows that students find the first two programming projects to be relatively easy and the last two programming projects to be quite difficult and time consuming. You are responsible for ensuring your programming assignments work properly on the `myth` machines.

Late Policy: The class material builds on itself and getting behind is taboo, since it tends to impede progress on the following assignments.

You get **four** free "late days" (24-hour periods) that you may use to extend the due dates of any assignment without penalty. Problem sets and assignments are always due at 4:00 p.m.

If you run out of freebies, you can still turn in your work late, but it will cost you. I deduct $3^{(1 + \text{days late})}\%$ from your grade on the assignment for each additional late day (e.g. 9%, 27%, 84%).

No credit will be given for any assignment not turned in with three days (72 hours) of the due date, regardless of late days.

Exams: Exams will open-book and open-lecture-notes, but you cannot use any electronic devices (including laptops). If this time does not work for you, then you will need to discuss this with me as soon as possible. Local SPCD students should come to campus for the exam if possible. The tentative exam schedule is as follows:

Midterm: Mon 21-July-2008 7-8:30pmPDT (Terman Auditorium)
Final: Friday or Saturday 15/16-August-2008 (time TBD)

Grading: This class is offered with both the graded and CR/NC options. The course grading is divided between the programming assignments and problem sets, a single midterm, and a final exam. The grade breakdown is:

Assignments	60 %
Midterm	15 %
Final	25 %

If you are taking the course CR/NC, you need to pull a C- or better in order to pass. All students must complete all assignments and exams, including those taking the course CR/NC.

Incompletes: You may only take an incomplete in the class if in case of a personal emergency (death of a close friend or family member, severe illness requiring you to remain in bed or in the hospital).

Partners: You can choose to do the assignments individually or work in pairs. If you work with a partner, you must hand in only one copy with both individuals' names on it. Both students receive the same grade in all circumstances. You can work individually on some assignments and with a partner on others. You may only collaborate with one partner for any particular assignment. Feel free to use the newsgroup (`su.class.cs143`) to find a partner if you want one.

Honor Code: You must complete written assignments on your own. You are permitted to discuss assignments *only* with the course staff and your partner. This includes both problem sets and programming projects. You may refer to any course material, textbooks, or material posted on the course website. You may not refer to past coursework unless explicitly provided by me (this means you cannot look at code written by past students!).

To be even more specific, you are not allowed to collaborate on the coding of your programs, nor are you allowed to copy assignments or even parts of assignments (you may refer to and use code from explicitly allowed references in the previous paragraph). The following three activities are among what I consider to be Honor Code violations in this course:

1. Looking at another's code or assignment.
2. Showing another student your code or assignment.
3. Discussing an assignment or code with someone other than your partner or a course staff member.

That said, you are permitted and encouraged to share test cases for your code with others. However, any sharing which is done must be done through the `shared_student_tests` repository (discussed in the first assignment's handout)