Compilers (Spring 2019 201930)

Instructor: **Lerner, Benjamin**
Subject: **CS**
Catalog & Section: **6410 01**
Course ID: **33587**
Objectives:

Enrollment: **37**
Responses Incl Declines: **25**
Declines: **0**

Course Related Questions (11 comments)

Q: Please comment on the strength and/or weakness of the required textbook/course materials.

1 The completion of a working compiler each week was integral to this course. I do not believe that I would have had nearly as much motivation or debugging ability otherwise.

2 There weren't a lot of materials available, and I got one of the "recommended" books which ended up being pretty unhelpful

3 this class *** rules

4 Amazing course website. Very thorough lectures with great examples. Would be nice for the "Do Now" examples in the lectures to either line up more with the homework, or to be completely disjoint with the homework.

5 Ben Lerner is not using traditional textbooks because he is taking his own approach to teaching Compiler Design. However, this is because he is (rightfully) unhappy with the way many textbooks teach the subject. His alternative seems like a much better approach, and I appreciate him going the hard-route and not following specific textbooks!

6 The assignments were really solid until tuples were introduced. Obviously the course is still being developed, but it was very frustrating it being in such an incomplete state.

The starter code given was often also incomplete, which is much more obnoxious.

The grading criteria isn't necessarily clear since there is not a suite of tests used by the graders. This should be prioritized.

7 Working without a textbook was fine, but it only worked well when the lecture notes were posted online. Towards the end of the semester, this stopped happening which was challenging.

8 Didn't really use the textbooks, but they contain useful information

9 The book was not really required. Class notes posted by professor were more than sufficient.

10 The lecture notes were generally good. There were some holes, but obviously they will improve after this semester. For the most part they were thorough and clear, and helped me understand a lot of concepts while working on the homeworks.

11 Lecture notes were a very helpful reference for assignments.

Learning Related Questions (15 comments)

Q: Please comment on the strengths of this course and/or ways to improve this course.

1 This class was largely experimental, which made it great at parts and confusing at others. Generally difficult information, and a huge time commitment outside of class, which is sad if you happen to be taking other difficult courses. I definitely learned a lot about compilers, but this course was much more low-level than I think I was initially anticipating.

2 This course is probably the most interesting course I've taken in my entire college career. It's fascinating to look under the hood of how programming language features that I've taken for granted are actually implemented and to implement a bunch of them myself.

Most of the lectures and assignments were well explained and laid out. Each assignment had us implementing another feature into our growing language. I really enjoyed seeing our language and our compiler grow and grow throughout the semester as it approached a pretty useful form.

While there was a substantial skeleton that our design had to attach to in the beginning, as the assignments progressed, we had more and more freedom to design our compilers how we wanted. However, for every assignment, we were also given some starter code for whatever feature we were implementing in that assignment. Adapting our heterogeneous designs to this new interface was often a multi-hour endeavor that had to happen before we even got to the assignment itself. If this was meant to be an intentional lesson, it was never explicitly stated in lecture or the assignments.

The written assignments were great dives into designing language features rather than implementing a design that was dictated to us. I'm more confident in my ability to implement a compiler than design one at this point; I would have liked some more design exercises, but I get that it's tough to balance giving students design freedom and to grade their assignments.

Grading of assignments was often pretty slow. We could submit two or even three assignments before the oldest one would get graded. This usually wasn't an issue, but it could be frustrating to not receive feedback before submitting the next assignment, especially in the later ones where we were implementing a substantial amount of complicated code.

Overall, I really liked this course. It demystified a lot of the "magic" of programming languages.

3 The course was mostly hands-on, as it should be.

One thing I found a bit frustrating was the delay between covering a piece of material in a lecture and using that material in the homeworks. For example, we would be covering material in class for homework 5, while still working on homework 4, which made it difficult to apply the learned material right away.

4 The only flaw in this course was the sheer volume of work. It was quite difficult to manage at times.

5 Love it, don't change a thing

6 Really great course, I like this method of iteratively developing increasingly complex compilers. Favorite course in Khoury college so far.

7 I learned a lot in this class and thoroughly enjoyed it. The only problem was that the homework was too time consuming. I don't want this review to deter anyone from taking the class, and this is just some feedback to help improve the course for the future.

1. Assembly knowledge was supposedly not required for this class. However, as someone who never programmed assembly in my life before this class, I will say that assembly knowledge is definitely very much helpful. In my opinion, we should have targeted our compiler to C instead.

2. We had to copy and paste our old code into the new starter code for each assignment. This theoretically sounds very easy, but in reality, it took at least a few hours to get all our old tests to pass due to implementation choice differences between the professor's code and ours. It was such a big difference that the final project took much less time than the past few assignments because we didn't have to do this step.

Q: Please comment on the strengths of this course and/or ways to improve this course.

8 Spread out the work load , it seemed too much in the beginning and middle of semester and really nothing in the ending month.

9 The material was very interesting, and the structure of starting with a basic functional compiler and expanding it, instead of going phase-by-phase was a fantastic approach.

Some of the assignments were disorganized and left us confused, but I think that will be ironed out more with each iteration.

Tehnically speaking, I think strong typing is more trouble than its worth in such a short semester. For us, it was good to learn for one assignment, but after that it got very hard to maintain and that effort detracted from us being able to work on and learn the newer features as much as we might have otherwise. In the future, I would recommend scrapping strong typing, introducing it after tuples and lambdas so that our original approach fits all those needs, or making strong typing be a one-off assignment before tuples, that we don't build off of.

Other than a few hiccups, the course was run very smoothly and I felt like I was constantly learning. Very well done.

10 I think that the format of this course was excellent and greatly helped me understand the importance of the different phases of a compiler. Some of the assignments were a little rough around the edges since Professor Lerner's version of this course is still new, but this wasn't too problematic and I think that things went pretty smoothly for the most part.

I think that I would have also appreciated a lecture about how our compiler architecture relates to what compilers for imperative languages do.

11 Hardest class I've taken but I still love it so much

12 Compilers is always going to be a difficult course with a large time requirement. But Ben's approach to the class made what should have been a hard class into a never-ending nightmare of constant work. But despite having an absolutely miserable semester in the course, I did learn a ton about compilers.

Specific issues:
* Many assignments in this course required an inhumane amount of work to be completed in a short period of time.
* Each (~weekly) assignment built off the prior one, but it took about a month for each assignment to be graded, so feedback often came too late to make a difference.
* Assignments were penalized for topics never discussed in class such as "ugly formatting" and inefficient code which seems unrelated to the core concept taught.
* The assignment release dates posted on the websites were occasionally outright lies where (as a hypothetical example) the website said an assignment was "Assigned Fri 02/01, Due Thu 02/07" but in reality the assignment was posted on the website on Sunday 2/3 or the starting assignment code was posed days later (both actual examples).

Other ideas for improvement:
* A full definition of the expected language (concrete syntax and all memory tags/layout) for each assignment would make things easier
* After each assignment is due, provide a test suite for the features from that assignment or a list of edge cases we may have forgotten to test so we both lose points for our mistakes but also fix them going forward.
* Keep the lectures better aligned with the homework material. By the end of the semester, we would be working on one assignment while learning about completely different topics we'd have to be using in the next assignment.
* Release all assignments in the same github repository so students can just git merge the new code in each week and better track bug-fixes to provided code vs the new assignments code instead of manually merging code each week.

Strengths:
* It was really cool to make a compiler that actually worked from the start of the course
* I learned far more than I was expecting to
* Ben's lecture style was great

13 Learning the material and completing the homework in just a week is usually an unreasonable task unless you choose to forgo any activities outside of schoolwork for the entire semester.

I highly recommend people take this class, but only if you already have a background in these topics, are okay with probably receiving a low grade, are okay with sacrificing sleep/social activities for a semester, or are taking a reduce courseload.

14 Overall, very good course. Like stated in class, it was the first iteration of the course, so a little here and there, but a good solid course that taught a lot of really good and interesting concepts.

I think the making the typechecker optional towards the end was a smart idea. Some of these concepts can get so complex, that even though it is realistic to maybe have to overhaul each phase to dd one new feature, having to do that in a week takes away from learning the basic concepts of that feature and pushes the assignment more towards debugging and phase interactions.

15 I really enjoyed this course and found it to be one of the most challenging I've ever taken. It was a lot more work than I think a class should be though (sometimes 30 hours/week). One thing I think could be improved was how we moved from assignment to assignment. I once spent three hours trying to merge our old code with the new starter code for the next one. Either assignments should be more like they are in PL where the solution of the previous assignment is given at the start of the new one, or the code should be more modularized (a separate file for ANF, well-formedness, compile, etc.) to make it easier to diff and add in new code.

Instructor Related Questions: Benjamin Lerner (14 comments)

Q: Describe instructor's strengths, areas for improvement, and any additional comments.

1 Ben is a great professor for this class; it's his area of research, and he clearly knows a lot about compilers, and has a lot of enthusiasm for them. His lectures were mostly really good, and he does a great job of clarifying points when you ask questions. The organization of the class as a whole could have been better thought out, but we were guinea pigs for his new curriculum so that's the way it goes.

2 Prof. Lerner is one of the best CS professors at Northeastern. I've taken several courses with him and he's always excited and effective while teaching. Since he focuses on programming languages in his research, he felt especially in his element.

3 World-class lecturer. Knows all the material, very honest and precise answers to questions (including the occasional "That's a good question, I don't know", which I think is very important). Lots of enthusiasm for the material, encourages class participation and creative answers. Made himself very available for extra help and discussion outside of class hours. Always responds promptly to questions on Piazza.

4 Great lectures as always

5 Always prepared, really great lecture notes, very knowledgeable. Would take pretty much any class with him.

6 Very good enthusiasm and obviously had a lot to say when it came to compiler design. No complaints on teaching style, all my complaints are course structure and materials related.

7 Ben is the only reason I took compilers and I do not regret that decision at all.

8 He is amazing, but may be too ambitious with the course.

9 Professor Lerner is one of the best CCIS has to offer. His lectures are also excellently planned and his style of lecturing is engaging and easy to follow. He does a fantastic job of introducing a problem then pushing the class towards a proper solution, with ample input from the students, as opposed to simply stating a problem and a solution, with little to no explanation why that solution is chosen. His lectures are very interactive, and students are always engaged with questions and ideas. He's always willing to pause the lecture to answer questions as needed. He clearly has a passion for the material he's teaching, it shows in his lectures, and as a result it makes it much easier to get engaged as a student.

10 Professor Lerner is very intelligent, a fantastic lecturer, and clearly has a lot of enthusiasm about compilers.

Q: Describe instructor's strengths, areas for improvement, and any additional comments.

11 feel the blern

also the TA Matt is pretty great

12 Ben is a great lecturer but could do better at listening to students and providing feedback during office hours.

The grading of our final projects seemed somewhat arbitrary and I was disappointed to see weeks of my work quickly evaluated in just 15 minutes based entirely on how I presented the project, not on how the code was written.

For my final project proposal, I laid out a system design and submitted it to Ben but got no feedback. Ultimately, during grading I was penalized for some of the design systems I described in my proposal which could've been avoided had we been given feedback on the proposal.

Ben used the time before class started for Q&A but it would have helped if that started at the beginning of class so those of us who couldn't get to class early could still hear the answers.

13 Great professor, knows the material really well and teaches it really well.

14 Prof Lerner is always prepared, knows what he's talking about and is always available to meet with students.