# Compilers (Fall 2019 202010)

Instructor: **Shivers III, Olin**<br>
Subject: **CS**<br>
Catalog & Section: **6410 01**<br>
Course ID: **16177**<br>
Objectives:

Enrollment: **21**<br>
Responses Incl Declines: **7**<br>
Declines: **0**

## Course Related Questions (5 comments)

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

1 The book could have been more comprehensive.

2 Textbook is ok. Some parts of the textbook are fairly confusing since it feels like it's asking you to design your compiler in a specific way without explaining to you how everything actually works. A big issue is being asked to stub functions where it's unclear where or how they're being used, making it confusing why they're being stubbed now rather than implemented later.

3 I successfully completed all the course assignments, and I expect to receive a high final grade in this course. However, I feel that my learning could have been much more efficient and effective with updated course material and streamlined course style.

The textbook is from the 1990s, using a language (Standard ML) that is not used today for any real purpose. It requires learning this language, along with several other intermediate languages, to complete the course. The textbook's techniques and explanations are out of date, and the book contains many errors and unnecessarily difficult/obscure ways of explaining what it is asking you to do.

The support code for the assignments is very lacking and incomplete, with far more busy-work required to understand how to work with it and to fill in the missing, uninteresting pieces that fill in the gaps than should be necessary. The same amount of actually-relevant material could be taught with far less busywork.

Additionally, there are no tests, auto-grading, or other framework for determining whether your code is correct. All code builds on previous assignments, so you can easily spend too much time on bugs caused by mistakes in prior assignments. There is no way to know whether you are doing everything correctly, and you don't get grades back in a timely fashion (at least an entire month between submitting an assignment and getting feedback).

We were never provided with a full syllabus. There was no provided grade breakdown, collaboration policy, course code of conduct, etc.

The lectures are performed on a whiteboard, with no slides or reference notes to look at later. If you have to miss a lecture, the textbook is the only way to get the information you missed.

4 The textbook was great, it explained the various topics well.

5 Strength : Helps build a compiler in a step by step manner, while also providing insight into important compiler technologies<br>
Weakness : Sometimes it seems informal due to missing/informal definitions.

## Learning Related Questions (3 comments)

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

1 There was a lot to learn in the subject. We had to skip some of the questions in the class because of time limitations, otherwise its perfect in all ways.

2 I did learn quite a bit about compilers in this course. I feel much more knowledgeable about the subject, and I think the course provided enough background information to give me a starting point for catching up with modern compilers research, if I need to. However, it was very inefficient and took too much time and work to teach its material. I am also not confident that it taught the most modern, relevant information.

This course is very out of date, inefficient, and over-stuffed with work. It should be updated to a modern programming language, modern compiler technology, and should provide helpful support code that is well-documented, along with tests and automated feedback for each assignment.

There are too many assignments, they take far too much time, and they are poorly defined and described. They should be streamlined and reduced to a reasonable workload. Additionally, the course currently has, but does not need, two exams on top of this large number of assignments. One of these two methods of evaluating learning should be cut or reduced.

3 As Olin mentioned, it is a capstone course with a hands on training and hence is invaluable for grad students. Improving the turnaround time for submitted assignments could help improve the class experience.

## Instructor Related Questions: Olin Shivers III (5 comments)

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

1 He is best and doesn't needs any improvement. He has in-depth knowledge of the subject and has excellent teaching skills

2 Professor is clearly very knowledgeable about the subject, and has a lot of interesting stories related to compilers or compiler history. Lectures are very useful and not just taught straight from the book. Professor cares a lot about the topic and spends time at the end of the course going over "bonus" content not required for the exam or project. Biggest issues with the course are that feedback is fairly slow for the assignments (since there is only one TA and assignments are complex), and there is very little material to prepare for the midterm and final (no practice exams or example questions).

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

3 The instructor describes concepts and algorithms in a very detailed, clear manner. He is knowledgeable about compilers, and he cares deeply about the subject. He shows his love of the material. He usually answers questions effectively and acts graciously when students correct mistakes. I was able to learn all the material effectively and expect to receive a high grade in the course.

However, the instructor is extremely prone to very long asides and unnecessary diatribes about information that is not directly relevant to the course. I tracked roughly how much time he spent on asides, and it often amounted to at least 30 minutes out of the 1.5 hour class periods. This made it very difficult to extract the useful information that might be relevant to the homework or to the exams from the irrelevant information.

The instructor tends to speak in ways that could be offputting and excluding to students from marginalized groups (i.e., not cisgender white men). Once, he intentionally misgendered a friend of his, who is a transgender woman, and claimed that she had given him "permission" to do so. In two classes in a row, he repeatedly asked the same student with a name that was not a traditionally white, American name to spell their name, and asked where they were from. He singled out another student to ask whether they spoke Mandarin, asking them about a particular Chinese phrase. He talked about Richard M. Stallman as a colleague who had done important work, without mentioning or acknowledging the serious allegations of abuse and misconduct that have been raised against Stallman.

The instructor's attitude is also elitist and would likely feel exclusive to students from a non-traditional computer science background. He often implied that everyone in the class should be people like him, who care deeply about every optimization and have a particular type of formal training in computer science. He often bragged about people he knew from the old days of computer science, and they were always men. He highlighted the accomplishments of these men, almost never mentioning women or people from marginalized groups. He used unnecessarily gendered language to describe computer science concepts, such as "this guy" to refer to variables or "brother" for relationships between objects. He brusquely interrupted students who were asking questions, assuming he already knew what they were asking.

At least twice, the instructor canceled office hours on the day-of, with only a verbal notification in class and no online notification. He often failed to respond in a timely fashion to questions on Piazza.

Grading was extremely delayed. The midterm grades were only available three days before the final, after the final class was over, and we had to visit his office hours or attend an optional, extra lecture to get the grades.

4 Prof. Shivers is a great teacher, he has a lot of personal experience with what he's teaching and he knows how to communicate it to students well. He makes the class interesting by telling anecdotes related to what we're learning, which prove the point that he's trying to make. He's also very helpful and accommodating if a student has a problem.

5 Olin is one of the best teachers I had opportunity to take a class with. His real life experiences make for interesting anecdotes/rants/arguments in the class. He provides interesting insights into how compiler technologies developed, who worked on what and the landscape of the area in 70s/80s, which I don't think any textbook could provide. I am very thankful to him for having introduced me to compilers in such an interesting way.