The University of North Carolina at Greensboro
CSC 439/639: Introduction to Compiler Design
Prof. Stephen R. Tate

Handout 1
August 19, 2020

CSC 439/639 Class Information and Syllabus

Instructor: Stephen R. Tate (Steve)
Lectures: Mon/Wed 2:00-3:15
Section 01A: Petty 224 on Mondays, Online on Wednesdays
Section 01B: Online on Mondays, Petty 224 on Wednesdays
Office: Petty 157
Office Hours: Mon/Wed 10:00-11:30 (or by appointment), virtual – see below
E-mail: `srtate@uncg.edu`

Note regarding in-person classes and meetings for Fall 2020: Due to the ongoing COVID-19 pandemic, in-person classes are very restricted, and office hours are online using the Zoom teleconferencing software. Students can connect during virtual office hours using the link provided in Canvas. All students are expected to be responsible regarding not only their own health, but also that of others. You may only be present in the classroom on days that you are authorized to be there. If you have any reason to believe that you might be sick, do not come to class and keep away from others until you know that you are not contagious. All classes are streamed live and are recorded for later viewing, and students will all have full access to all resources whether they are attending classes in person or participating remotely. If on-campus class sessions are canceled for any reason, including worsening general situation or instructor health, class will go on! Make sure you stay on top of your email and Canvas announcements and be prepared to be flexible and responsive to changes. More information COVID-specific class protections and policies is at the end of the syllabus.

Class Web Page: `http://www.uncg.edu/cmp/faculty/srtate/439.f20/`

Catalog description: Basic techniques of compiler design and implementation: lexical analysis, parsing, code generation. Sizable programming project implementing a compiler for a block-structured language with strong typing.

Prerequisites: Grade of C or better in CSC 261 and CSC 330, or permission of instructor. *Note that having previously taken CSC 452/652 (Theory of Computation) is helpful.*

Longer Description: This class provides an introduction to how compilers and language translation systems work, including both the underlying theory and the practical implementation. This is a programming-intensive class, and by the end of the class each student is expected to have produced a fully-functional compiler, which will be thousands of lines of code. You *must* be comfortable and confident in your ability to crank out non-trivial amounts of code to succeed in this class.

Student Learning Outcomes: Upon successful completion of this course students should be able to

1. Describe the basic components of a compiler and how they inter-operate;

2. Solve written problems involving theory and practice of lexical analysis and state machines;
3. Implement a lexical analyzer;
4. Solve written problems involving theory and practice of parsing and context-free grammars;
5. Implement a parser for a given grammar;
6. Solve written problems involving theory and practice of code generation (both intermediate code and native code), including some basic optimization problems;
7. Implement the code generation phase of a compiler;
8. (Graduate Students) Demonstrate familiarity with current research in language translation and analysis, *or* implement an advanced compiler stage/project.

Textbook and Readings: The required textbook is

> Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools, second edition*, Pearson, 2007. ISBN-13 978-0-321-48681-3.
>
> A textbook errata sheet is available from the authors.

Additional readings may be assigned throughout the semester, and will either be freely available or copies will be provided for students.

Topics: This course is only sporadically offered, and so the topic list and timings are not as well-developed as some other courses. The following is a list of planned topics, and an estimate of the number of days devoted to each. A precise schedule, updated as the semester progresses, is on the class web site.

Class Overview and Introduction to Compilers (Chapter 1) [2 days]
A Simple Syntax-Directed Translator (Chapter 2) [2 days]
Lexical Analysis (Chapter 3) [4 days]
Syntax Analysis (Chapter 4) [5 days]
Syntax-Directed Translation (Chapter 5) [2 days]
Intermediate-Code Generation (Chapter 6) [3 days]
Run-Time Environments (Chapter 7) [3 days]
Code Generation (Chapter 8) [3 days]
Basic Optimizations (part of Chapter 9?) [1 day]

Teaching Methods and Assignments: This class will meet for two 75-minute periods per week, and class meetings will consist of a combination of lecture/presentation, discussion, and in-class exercises. The meeting format will depend on the current COVID-19 pandemic situation, and if we are not all able to meet in person then lectures will be available via a live video stream and as a recorded lecture (both available in Canvas). Students are expected to be prepared and actively participate in class, having done all required readings in advance. Grades are based on student work done in homework assignments, exams, and a large multi-stage programming project (a compiler).

Homework Assignments: Weekly homework will use the online Gradiance system, and is due every Friday, with a few exceptions. These assignments are designed to keep you on schedule with readings and topics, so will not be accepted late, but your lowest grade during the semester will be dropped. Information on signing up and using this system will be provided in class. The Gradiance system is tightly coordinated with the textbook material and your assigned readings, and takes a somewhat unique approach: Questions typically involve working out the solution to a problem, and then answering multiple choice questions that are asked about *random parts* of the solution. You may attempt to answer the homework questions as many times as you want, but there is a 10-minute time delay between attempts and answer choices are randomized with each attempt in order to discourage repeated random guessing. The way the problems *should* be approached is to work out the core problem fully so that you can then answer the multiple choice questions easily – don't approach it from the beginning as a multiple choice question! I will happily give assistance on these homeworks, but my first question will always be "show me your full solution" – I will not help you approach the problem as a multiple choice question. The system gives you immediate feedback on your answers, with incorrect answers giving clues as to where you might have gone wrong in your larger solution. Your grade on each homework is the highest grade of any attempt (*not* the last attempt, so if you want to go back and explore other possible answers you can do that without the possibility of lowering your grade). If you start in plenty of time, then even with the time delay between attempts every student should be able to get a perfect score on the homework every week.

Programming Project: The most significant student-work for this class will be the implementation of a compiler, assigned in five stages throughout the semester. The final stage, completing a fully-functional compiler, is due at beginning of the university-scheduled final exam (Monday, November 30, 2020, at 3:30PM). This project takes the place of a final comprehensive exam. Programming projects will be managed and turned in using GitHub – specific information will be provided in class, but students who do not currently have a GitHub account are encouraged to set one up as soon as possible, and to sign up for the free student education pack. Programming projects are individual work, and collaboration and sharing of code is not allowed, so all GitHub repositories must be private (they will be set up this way be default by GitHub classroom, so do not change the defaults). Students *are* allowed (and encouraged!) to create and share test cases for all stages of the project, and a shared repository will be provided for easy sharing of these test cases. Except for the final assignment (the complete compiler), programming assignments may be submitted up to one week late, with a late penalty (see below). The final compiler will not be accepted late.

Exams: There will be two mid-term exams in this class. Note that the second exam is on the final class day (Tuesday, November 24), but will be a regular (non-comprehensive) midterm exam. There is no written final exam, but instead a final project will be due at the university-scheduled final exam time. COVID-19 adds significant uncertainty to the structure and taking of these midterm exams. Ideally, these exams will be taken on campus by all students at the same time, which means that students may need to plan to come to campus on what would normally be a "remote day." Depending on final class enrollment, we will reserve either a larger room or multiple rooms so that all students can be accommodated while maintaining required social distancing. If necessary, we will consider remote testing or an online exam, but this will only be done as a last resort.

Graduate Students: Graduate students taking this class as CSC 639 (and receiving graduate credit) will be required to complete an additional, advanced project, related to learning outcome 8. This can be either a report on current language/compiler research, or it can be an implementation project that either (a) creates a more advanced optimization stage for your compiler or (b) performs real-world code-analysis using a commercial-grade compiler infrastructure (LLVM). More information and guidance will be provided for this project part-way through the semester.

Evaluation and Grading: Each student work product will be graded, and the student's final grade will be determined by assigning each category of work a weighted score according to the distribution below, and then the final weighted average is mapped to a letter grade as shown.

For undergraduates:

Category	
Compiler Project	50%
Homeworks	20%
Mid-term Exam 1	15%
Mid-term Exam 2	15%

Letter Grade Assignment				
	[87.5 , 89.5) = B+	[77.5 , 79.5) = C+	[67.5 , 69.5) = D+	[0 , 59.5) = F
[91.5 , ∞) = A	[81.5 , 87.5) = B	[71.5 , 77.5) = C	[61.5 , 67.5) = D	
[89.5 , 91.5) = A-	[79.5 , 81.5) = B-	[69.5 , 71.5) = C-	[59.5 , 61.5) = D-	

For graduate students:

Category	
Compiler Project	45%
Homeworks	15%
Graduate Project	10%
Mid-term Exam 1	15%
Mid-term Exam 2	15%

Letter Grade Assignment			
	[87.5 , 89.5) = B+	[77.5 , 79.5) = C+	[0 , 71.5) = F
[91.5 ,) = A	[81.5 , 87.5) = B	[71.5 , 77.5) = C	
[89.5 , 91.5) = A-	[79.5 , 81.5) = B-		

Academic Integrity: Students are expected to be familiar with and abide by the UNCG Academic Integrity Policy, which is online at `http://academicintegrity.uncg.edu/`

Assignments in this class are for individual work, unless explicitly stated otherwise. General concepts and material covered in the class may be discussed with other students or in study groups, but specific assigned problems should not be discussed and all submitted work should be entirely your own. If you use external references (including web sites, books, etc.) in preparing your solutions, you should clearly mark the part(s) of your solution influenced by these references and provide clear citations to the source of information you are using. Sharing your own work is a serious violation of academic integrity, and if homework is copied then *both* the person who actually did the work and the person who copied it will be punished. Any incidents of academic dishonesty will be handled strictly, resulting in either a zero on the assignment or an F in the class, depending on the severity of the incident, and incidents will be reported to the UNCG Office of Student Rights and Responsibilities.

Attendance Policy: Students are responsible for everything said or done in class, including material that is not in the book or readings, and information regarding assignments and due dates. That said, attendance is not required, and in-person attendance may be difficult due to COVID-19 conditions. All classes will be streamed and recorded, however, so students are expected to either attend in person, participate during the live feed, or watch the recorded class within 24 hours of the scheduled class time. Students with planned schedule conflicts, whether due to religious observances or other reasons, must inform the instructor in advance and if necessary to turn in any work early so that deadlines are met.

Late Policy and Makeup Exams: As stated above, homework assignments are due each Friday, and since they are designed to keep you on schedule with class topics they will not be accepted late. Programming assignments other than the final submission (the complete compiler) may be turned in up to 7 calendar days late with a 25% late penalty. The final compiler will not be accepted late. Students with planned absences, whether for university events, religious observance, or other reason, are expected to make arrangements with the instructor to turn in assignments or take exams before the scheduled date of the assignment or test. No assignment will be accepted more than 7 calendar days after the original due date!

Exam/test dates will be announced at least two weeks in advance, and may be made up only if it was missed due to an extreme emergency and arrangements are made before the exam date. Exams may not be taken early or late due to personal travel plans.

Given the COVID-19 situation, I will be flexible and accommodating within reason, but students *must* inform me of any complications in advance of due dates.

In-class Behavior: When you are in class (in-person or remotely) you should be focused on the class, and you should act in a professional and mature manner, avoiding behavior that is distracting to the learning environment of other students. If you are participating remotely, you should keep your microphone muted when you are not actively engaged in a class discussion. To promote a sense of community, you are asked to turn on your camera when asking or answering a question remotely.

ADA Statement: UNCG seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Office of Accessibility Resources and Services located in 215 Elliott University Center: (336) 334-5440 (or on the web at http://oars.uncg.edu).

University COVID-19 Policy: As UNCG returns to face-to-face course offerings in fall 2020, the campus community must recognize and address concerns about physical and emotional safety. As such, all students, faculty, and staff are required to uphold UNCG's culture of care by actively engaging in behaviors that limit the spread of COVID-19. Such actions include, but are not limited to, the following:

- Wearing a face covering that covers both nose and mouth
- Observing social distance in the classroom
- Engaging in proper hand washing hygiene when possible
- Self-monitoring for symptoms of COVID-19
- Staying home if you are ill
- Complying with directions from health care providers or public health officials to quarantine or isolate if ill or exposed to someone who is ill

Instructors will have seating charts for their classes. These are important for maintaining appropriate social distance during class and facilitating contact tracing should there be a confirmed case of COVID-19. Students must sit in their assigned seat at every class meeting and must not move furniture. Students should not eat or drink during class time.

A limited number of disposable masks will be available in classrooms for students who have forgotten theirs. Face coverings will also be available for purchase in the UNCG Campus Bookstore. Students who do not follow masking and social distancing requirements will be asked to put on a face covering or leave the classroom to retrieve one and only return when they follow these basic requirements to uphold standards of safety and care for the UNCG community. Once students have a face covering, they are permitted to re-enter a class already in progress. Repeated issues may result in conduct action. The course policies regarding attendance and academics remain in effect for partial or full absence from class due to lack of adherence with face covering and social distancing requirements.

For instances where the Office of Accessibility Resources and Services (OARS) has granted accommodations regarding wearing face coverings, students should contact their instructors to develop appropriate alternatives to class participation and/or activities as needed. Instructors or the student may also contact OARS (336.334.5440) who, in consultation with Student Health Services, will review requests for accommodations.

Health and well-being impact learning and academic success. Throughout your time in the university, you may experience a range of concerns that can cause barriers to your academic success. These might include illnesses, strained relationships, anxiety, high levels of stress, alcohol or drug problems, feeling down, or loss of motivation. Student Health Services and the Counseling Center can help with these or other issues you may experience. You can learn about the free, confidential mental health services available on campus by calling 336-334-5874, visiting the website at https://shs.uncg.edu/ or visiting the Anna M. Gove Student Health Center at 107 Gray Drive. For undergraduate or graduate students in recovery from alcohol and other drug addiction, the Spartan Recovery Program (SRP) offers recovery support services. You can learn more about recovery and recovery support services by visiting `https://shs.uncg.edu/srp` or reaching out to recovery@uncg.edu

COVID-19 Spartan Shield Video: UNCG Chancellor Frank Gilliam has challenged us to create a Culture of Care at UNCG where we all wear face coverings and social distance, less to protect ourselves but rather more to protect everyone around us. It shows that you care about the well being of everyone around you. We have created this video featuring your student body presidents to better explain how and why this is so important.

Please watch this video before the first day of classes: `https://youtu.be/Mb58551qxEk`