

*ECE466 Compilers Syllabus -- Spring 2025 -- Prof. Jeff Hakner*

*Overview*

This is a graduate-level course in Compilers. Although the title of the course is "compilers" we will also be discussing interpreters and related concepts. In many CS curricula, much of the material that I present in this course would actually be in a 300-level undergrad course, and some of the material might be in a followup grad-level course. In our curriculum, we only have this one course in compilers. Therefore, ECE466 must serve as an introduction to compilers, as well as a taste of graduate-level topics in compilers.

*Student Expectations*

Because this is being run at the graduate level, it is important for students to understand the expectations, which differ from undergrad courses. There is very little "hand-holding" in this course. Students are expected to do a substantial amount of independent learning. The "lecture notes" do not comprise the entirety of the material for which the student is responsible.

This is a capstone project-based course in which we'll be (collectively, yet individually) building a rudimentary compiler. Therefore the individual homework assignments are really milestones in the journey to the project's completion. It should also become apparent that each assignment builds upon the previous assignments.

Therefore I make this critical recommendation: **Do not fall behind!** The complexity of the work (both intellectual complexity and technical complexity) increases as we go forward. A delay of say a single week in completing the first assignment might seem inconsequential in a 15 week semester, but it will put you in the bad position of being unable to start on the second assignment while the rest of the class is moving ahead, and so on, as we move through the project.

I would also expect that students read the chapter material ahead of the lectures. You will be in a better position to ask questions during lecture if you are already somewhat familiar with what is being discussed.

*Classroom Environment*

Please do not use devices such as cell phones, laptops, tablets, etc. during class except where specifically requested by me (e.g. to display code to the rest of the class). These devices should be turned off and stowed out of sight during class time.

As a grad-level elective class, attendance is optional. Of course I believe there is significant value in attending the lectures, and I do place some weight on class participation in determining the final grade. However, there is no need to request my permission in advance to be absent from a single class. If you will be absent for multiple weeks in a row, it is best to let me know so we can work out what impact that might have on your progress.

From time to time, we will have project progress reviews where some or all students/groups present their current work. The easiest way to do this in class is for students to use the "Screenbeam" aka "wifi screencasting" feature. If that is not possible, you would have to come to the podium and connect up to the HDMI input, which wastes time and can be troublesome. Please, prior to these code reviews, make sure you are able to use the Screenbeam feature to

demonstrate.

### *Major Objectives*

In ECE466, we have the following broad learning objectives:

- \* How a compiler and interpreter work
- \* Using specialized compiler programming tools such as Flex and Bison
- \* Gaining a more detailed understand of a sample language (we'll be using C)
- \* Assembly Language
- \* Run-time environment and Application Binary Interface
- \* How the linker/loader and other OS-dependent tools work with the compiler

### *Office Hours / Contact*

The best way to reach me with a question is via email to [hak@cooper.edu](mailto:hak@cooper.edu). I will try to answer student email questions within 12 hours, including weekends. If we need to, we can schedule a one-on-one TEAMS meeting.

I am also generally on campus Mondays, Tuesdays, Wednesdays in Room 810.

Find lecture notes, assignments, and supporting material at  
<http://faculty.cooper.edu/hak/ece466/>

### *Grading & Group Work*

In a graduate course, it is very unusual for grades other than A or B to be given out. I will base the grade primarily on your submitted project work. We will also have a midterm quiz and final quiz, dates and modes TBD. When a grade falls on the borderline, class participation and other soft factors are considered.

Therefore, in the context of ECE466, "A" will be earned if the project is completed successfully and all other assessments are satisfactory. "B" will be given out for partial project completion. "C" is reserved for a disaster such as minimal progress on the project, or significant problems with the assessments.

Partnered work is permitted. In other words, you may work alone, or you may work in a group with one other student. This is only for the project component of the course. Other assessments continue to be on an individual basis. Some students may prefer to work alone. There is no bias either way in terms of grading. Note that since the project is cumulative, a decision to form a group should be made during the first week. Any decision to split up a group after work has already been submitted must be mutually agreed by both group members. Likewise, forming a group beyond the first assignment submission may mean one student having to discard their work, and is thus counter-productive. I am not able to mediate any disputes between partners.

### *Accommodations*

Please consult the following web site:

<https://cooper.edu/students/student-affairs/disability>  
to learn more about Cooper Union's commitment to a classroom environment which is equitable for students with disabilities or diagnosed learning differences. If you have a letter of accommodation provided by the Office of Student Care, it is very important that we schedule a private one-on-one meeting as early as possible in the semester (or when the letter is provided, if mid-semester) to discuss how we can accommodate your needs consistent with the educational goals of the course. We must have an agreed-upon plan in advance. Per school policy, accommodations are not retroactive. In general, accommodations of extra time will not be applicable in this course, since as previously mentioned, all project deadlines are really milestones toward the completed capstone project. We will not be able to extend when that final project is due because of school deadlines for submission of grades.

### *Academic Honesty*

The project work is expected to be your own original, individual (or group, if working with a partner) intellectual work. Of course, you will obtain ideas, algorithms, data structures, and even small passages of code from other sources. These must be attributed via comments in your source code.

### *Textbook*

There is no required textbook for this class. I suggest a number of useful references on the course website.

Aho, Alfred V., Lam, Monica S., Sethi, Ravi and Ullman, Jeffrey D.: *Compilers: Principles, Techniques, and Tools*. 2nd edition. Addison-Wesley, 2006.

The first edition of the textbook is also quite workable, especially for the earlier units of this course.

Students might find the O'Reilly book on Lex and Yacc (two tools which are used to automate the front-end of the compiler) helpful. However, much of the information in this book is dated and available for free online or in the man pages, which students should absolutely read thoroughly!

The make program will prove very useful in developing the compiler. There is an O'Reilly book on make, but again for the purposes of this project, there is enough free information available online.

Harbison, Samuel P and Steele, Guy L.: *C: A reference manual*. 5th edition. Prentice Hall, 2006.

This is basically a regurgitation of the ISO C-99 standard, but with better examples and explanations, and a discussion of backwards compatibility issues with ANSI C (1989 standard) and classic K&R C. While it is now dated (there was never a 6th edition covering more recent developments) it provides an excellent alternative analysis of the actual C standard (which can be very confusing at times)

On the course web site, I give links for various versions of the C language standard, and several assembly language references. You are also free to seek out your own reference sources.

### *Course Outline*

Below is the intended list of topics and the time allotted to each. This is subject to change depending on how the class progresses and scheduling issues. There are also approximate project milestones along with what percentage of the total work on the project they each represent and guidelines for when they should be completed. Again, subject to change.

Unit 1: Introduction to Compilers & Interpreters, Lexical Analysis  
(milestone 1: lexer, target 2/5, 5% of total project)

Unit 2: Parsing (syntactical analysis)  
(milestone 2: first parser, target 2/19, 10% of total project)

Unit 3: Error handling, memory allocation, other misc. issues

Unit 4: Semantic Analysis, symbol tables, type systems  
(milestone 3: declaration parsing/symbol table/type representations)  
target 3/12, 20% of total project)

*NOTE: Wed 3/12: Class does not meet (Monday schedule)*

*NOTE: Tue 3/18: Last day to withdraw from a course.* It is rare for students to withdraw from ECE466, but you should consider it if by this point you are not close to completing the milestone which was due on 3/12

Unit 5: Intermediate Representations  
(milestone 4: Complete parser / AST, target 4/2, 15% of total project)  
  
(milestone 5: generation of IR, target 4/23, 35% of total project)

*NOTE: Wed 4/16: Spring Break*

Unit 6: Architecture-Neutral Optimization

Unit 7: The Target Environment / Assembly Language

Unit 8: Target Code Generation  
(milestone 6: completion of the compiler to assembly language,  
ABSOLUTE DEADLINE 5/15, 15% of total project)

*Wed 5/14: Final quiz and in-class review of final project progress*

*Senior grades are due 5/19. All work must be completed by 5/15 so that grades can be completed on time.*