Syllabus

Summary

This course is designed to teach you practical programming skills. Everything you learn here will be widely applicable to academic and industrial work.

After taking this course, you will:

- Be capable of using a terminal
- Be capable of using Bash and writing scripts
- Be capable of writing C code
- Be capable of writing Python code
- Be prepared for future courses that require programming
- Be a more self sufficient programmer

Lessons

Some of these will take multiple classes to fully cover. Assignments may follow lessons or the completion of an entire topic.

The course specifically requires covering linux and C, though there are more topics we will cover if time permits. The course requires the C and Python topics, but time permitting we'll cover a few more languages:

0. Course Overview
1. Introduction to the Terminal, Bash, and Make
2. C: Compiling, Assembling, Linking, and Executing
3. C: Types, Operators, and Expressions
4. C: Control Flow
5. C: Pointers and Arrays
6. C: File IO
7. C: Structures, Typedefs, and Basic data structures
8. C++: Object Oriented Programming Basics
9. Python: Basics
10. SQL: Basics
11. HTML: Basics

Textbooks

Optional: *The C Programming Language* by Kernighan and Ritchie.

Grading

- Homework Assignments 70%
- Quizzes 30%

Almost every lesson will be supplemented with an assignment. Each assignment will either be assigned individually or in small groups. Assignments make up a very large portion of you grade so make sure you put in the effort. All assignments must be submitted one week after they are assigned **before** the start of class. My class begins at 6 pm, that means assignments are due at 5:59 pm eastern time. This is my policy because I know how many students will work on the assignment during class otherwise. A late assignment receives a zero.

I do however, offer something almost no other professor will offer you: 2 assignment

re-submits for the semester. You will be allowed to make revisions for 1 week after receiving the grade for full credit. This policy is a reflection of the fact that no code is perfect on the first try. Some assignments will have automated tests that can tell you whether your code produces the expected output.
The grading breakdown for an assignment is roughly as follows:

Correctness: 85% Style: 15%

What I mean by correctness should be obvious: inputs produce expected outputs. Style is slightly more subjective. Mainly it means your code is easily readable and well organized. We will discuss specific expectations later.

If your program does not run at all, the grade is 0 no matter what.

Assignment Submission

Assignments in past years by other CS102 faculty have used Github education. I've heard many students have had trouble with the submission process, so I'm doing email submissions instead. Please send submission emails with subject: "CS102 [first name] [last name] Assignment #X" to ross.kaplan@cooper.edu.

Quizzes

Quizzes will be used as a way for me to gauge the understanding of the class, so it is important that everyone try their best. It is not my way of punishing you for not understanding.

Office Hours

I will hold office hours right after class. Office hours a great time to ask in-depth questions that couldn't normally be asked in class due to time constraints. This includes help with assignments and help fixing your programs. I am also available for personal appointments outside of business hours, just message me to set one up. Lastly, I'm happy to help by email as well!

Academic Honesty

If you use code from another source (a website, a book, or another student) you must cite it with a comment. If you do not cite it, you will receive zero credit for that assignment. You should never copy another student's code directly. You may ask for advice or single-line snippets but if I find that significant portions of your assignment have been copied from another student, you will receive a zero, and I will be obligated to report it.

The above does not apply to LLM-generated code. Do not use chatgpt, bard, or any other AI-based code tools that generate code for you. Even cited, LLM code generation is strictly off-limits.

If you have *any* questions about the academic honesty policies of this course, please ask me.