Class Overview

My Introduction

My name is Ross Kaplan, you can call me Ross, Mr. Kaplan, or Professor Kaplan, whatever you are comfortable with. This is my first time teaching this class.

I graduated from The Cooper Union in 2018 with a bachelors degree and in 2019 with my masters degree, both in electrical engineering. Since then, I've worked at Goldman Sachs as a software engineer on core infrastructure and later on a trading platform on backend infra. I currently work at Google on mobile data infrastructure, building software for mobile devices and google servers for synchronizing both public and private data.

Class Introduction

You probably don't need an explanation of computing, since it is now everywhere. In short though, computing is just the carrying out of specific instructions. This is embedded in a huge range of applications including:

- building a web application
- simulating physical structures
- controlling a nuclear reactor
- automatically controlling drones
- beating humans at chess, go, or league
- precisely landing rockets
- determining protein foldings

You probably won't be able to do those things right after finishing this class, but you will have the basic foundations to start to understand how they could be done.

The class will focus very much on practical skills that go into programming. We will go through multiple languages, each with a different purpose, and we will practice them by writing interesting programs. We will also learn some tools standard to programmers, such as Bash, and time-permitting by the end of the semester, maybe even a little bit of theory, such as time complexity analysis.

After this class, you will be able to write your own programs, and learn more languages and tools on your own. You will understand the elementary components that comprise most programs. You will have a greater perspective on how the digital world around you works as whole.

This class is also a prerequisite for ECE264, or Data Structures and Algorithms 1, which is a gateway to most other CS courses at Cooper, including getting a CS minor. If you enjoy the topics covered in this course, I highly recommend taking that course as well.

Attendance

Attendance is not mandatory, but I highly suggest it. If you don't want to attend classes, you may be better off watching videos and reading tutorials online rather than taking this class. I'm not kidding! That is still a really good way to learn computer science. Some great online lectures include *CS50* from Stanford and *The Missing Semester* from MIT.

The benefits you get from actually attending class are:

- live interaction with your peer group, a shared experience.
- personalized teaching; and
- motivation to actually pay attention and learn.

I will still expect you take the quizzes however, regardless of if you attend the full lecture or not. If you have extenuating circumstances, please communicate with me, I am very flexible. However, without communicating, missing the quiz will result in a 0.

Academic Honesty

You can use look up small snippets of code online, but it is never appropriate to copy code verbatim without understanding it. If you copy entire assignments or sizable chunks from the internet or classmates, that is cheating. You can certainly ask for advice, however. If you do end up copying a small amount of code from a classmate or online source, I ask that you:

1. Type it out manually and think about what you are writing rather than just copy pasting the whole thing.
2. Attribute the source in a comment.

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 I find you are cheating on an assignment you will receive zero credit, and I am obligated to report the incident.

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

Topics

This course is not designed to give you a particularly deep introduction into any field of computer science, but a broad overview of many parts. We begin with a very "low level" language, C which will help you understand some of the inner workings of a computer. We will then briefly touch on C++ to give you an introduction to "object oriented programming". Then we will move onto Python, a high level language that you can use for fast prototyping. Time permitting, we will then spend some time with SQL (Structured Query Language) which is the de-facto language for interfacing with databases and possibly cover some web development.

If you want to delve into these subjects in a deeper way, Cooper Union offers these courses through the Electrical Engineering department:

- Hardware: Electronics, Digital Logic Design, Computer Architecture
- Sensors: Digital Signal Processing, Control Theory
- Networking: Computer Networking, Communication Theory
- Low level programming: Operating Systems, Compilers, Computer Security
- Advanced Algorithms: Data Structures and Algorithms, Machine Learning
- Databases, SQL: Databases
- Web Development: Large Software Projects

If any of these things interest you, speak to me, or some member of the Electrical Engineering faculty as soon as possible and we can help you figure out if the classes are right for you and what pre-requisites you may need.

Strategies for Learning

The same as any other skill, the best way to learn coding is to do it. If you simply write code frequently with the intention of making something interesting or useful, you will learn to code. Whenever you write code, you should always be thinking of how to make it better. For many problems you encounter, you should always feel comfortable searching for the solution on the internet. One very valuable thing to learn is how to search for the solution, and how to describe the problem in a way that is can be easily searched. We will work on building coding vocabulary throughout the class.

Questions

I will be asking all of you for a steady stream of feedback to make sure that my teaching is working for you. I come from a place of a decent amount of experience, so it can be difficult for me to realize when something is not obvious, or that some students might be lacking some requisite knowledge.

If you find something confusing, please call me out immediately as it is likely that other students feel the same. This class is for beginners, not people who already have programming experience. Therefore, no question you could possibly ask that is too basic for me to answer. If you have any question during class just raise your hand. If you are to shy for that, feel free to message me privately.

If you have questions outside of contact hours, just send me an email at ross.kaplan@cooper.edu.