

## Course Placement Information

---

*Based on a handout by Eric Roberts and Mehran Sahami.  
Thanks to Mehran Sahami, Cynthia Lee, Nick Parlante, and Patrick Young for their input!*

The CS department offers many programming courses and it can be difficult to know exactly where to begin. This handout provides a general overview of these classes and some advice about selecting the class that's right for you.

If you have any questions, please feel free to come talk to the course staff! We'd be happy to offer one-on-one advice.

## Getting Started in CS: CS106A

CS106A is our first-quarter programming course. It teaches the widely-used Python programming language along with good software engineering principles. If you're interested in learning how to program a computer, this is the place to start. CS106A has no prerequisites – it's open to everyone! The course is explicitly designed to appeal to humanists and social scientists as well as hard-core techies. In fact, most CS 106A graduates end up majoring outside of the School of Engineering. The course requires no previous background in programming, but does require considerable dedication and hard work.

We've found that some high school CS classes are more rigorous than others. If you do have high school CS experience, we recommend reviewing the syllabus for CS106A to get a sense of what it covers. If you feel that you've seen all these topics before, you may want to start off in CS106B. As always, feel free to ask the course staff if you have questions about where to begin!

We recommend that you take CS106A if

- You are interested in learning to program computers.
- You have little to no prior programming experience.

We recommend that you *not* take CS106A if

- You have prior programming experience at a level comparable to an introductory college course (for example, if you scored a 4 or 5 on the AP CS exam).
- You have prior programming experience in some other programming language and just want to learn how to program in Python.

In the 2020-2021 academic year, CS106A is offered every quarter.

## The Next Step: CS106B

CS106B is our second course in computer programming. It focuses on techniques for solving more complex problems than those covered in CS106A and for analyzing program efficiency. Specifically, it explores fundamental data types and data structures, recursive problem solving, graph algorithms, and basic algorithmic analysis.

CS106B assumes programming experience at the level of CS106A, though you don't necessarily have to have taken CS106A to take CS106B. If you've seen basic control structures (loops, `if` statements, etc.), variables, arrays, and program decomposition, then you should be ready to take CS106B.

CS106B is taught using the C++ programming language, but uses C++ more as a vehicle for ideas than as an object of study. If you're interested in studying C++ in depth, consider looking into CS106L.

We recommend that you take CS106B if

- You have prior programming experience at the level of CS106A.
- You are interested in learning more about general computer problem-solving.
- You've programmed before but have not seen recursion, data structures, or algorithmic analysis.

We recommend that you *not* take CS106B if

- You already have completed equivalent coursework elsewhere.
- You have little to no prior programming experience.
- You have prior programming experience in some other programming language and just want to learn how to program in C++.

In the 2020-2021 academic year, CS106B is offered every quarter.

## Optional Add-Ons to CS106B

We offer three courses that are designed to complement CS106B with additional material. *None of these courses are required*, and they do not count toward the CS major or CS minor requirements. However, if you're interested in going deeper for your own enrichment, you may find them worthwhile!

### *Exploring More: CS106M*

CS106M is an add-on course to CS106B that explores applications of the concepts to more advanced areas of computer science. For example, this year's offering will likely cover topics like data compression, error-correcting codes, and digital signatures. The topics covered in CS106M will not be required by later CS courses, even if you are planning to major in CS.

We recommend that you take CS106M in addition to CS106B if

- You want to see applications of CS106B concepts to more specialized areas in CS.
- You are willing to put in more work than is necessary for CS106B.

We recommend that you *not* take CS106M if

- You are concerned that you “need” to take CS106M to avoid falling behind everyone else (trust us, that's not going to happen.)

In the 2020-2021 academic year, CS106M is offered in Fall quarter.

### *Coding for Good: CS106S*

CS106S is an optional 1 unit add-on course to CS106B that gives you a chance to work on programs for social good. The class brings in student groups, nonprofits, and local tech companies and is a mix of a speaker series and small project course. The course also teaches basic web development, but is not meant to be a stand-alone web development course.

We recommend that you take CS106S in addition to CS106B if

- You are interested in exploring social good applications of computer science.
- You are willing to put in more work than is necessary for CS106B.

In the 2020-2021 academic year, CS106S is offered year-round.

### *Learning a Language: CS106L*

CS106L is a one-unit companion course to CS106B that focuses purely on the C++ programming language. Unlike CS106A and CS106B, which focus more on general programming skills and fundamental programming concepts, CS106L is specifically designed to focus on language features particular to C++ and how to use the C++ programming language to solve problems. Although CS106L is designed as a companion course to CS106B, it's open to anyone with a comparable background.

We recommend that you take CS106L in addition to CS106B if

- You are interested in learning more about the C++ programming language.

We recommend that you *not* take CS106L if

- You want a deeper understanding of topics like recursion, data structures, or big-O notation.

In the 2020-2021 academic year, CS106L is offered year-round.

## A “General Education” Introductory Course: CS105

If you're looking to learn more about computers or computing (for example, you want to learn how to make a web page, or how the Internet works, or how a computer is put together) without focusing too much on programming, you may want to try CS105. This course is designed to give a broader introduction to computing than CS106A at the expense of more significant programming depth. It attracts an audience of approximately 150 students a year, most of whom take the course primarily to meet the “Formal Reasoning” Ways of Thinking/Doing requirement. If your only interest is in meeting that requirement, CS 105 is likely to be the most appropriate course. Note that CS105 does not lead to CS106B. If you took CS105 and decided you wanted to continue taking CS courses, you would still need to take CS106A. Like any programming courses, CS 105 requires a reasonable amount of work, but not as much as CS106A.

We recommend that you take CS105 if

- You have no prior background in computer programming.
- You are interested in learning about computers and how programs control computers.
- You are interested in exploring programming in less depth than what's covered in CS106A.

We recommend that you *not* take CS105 if

- You know for certain that later on, you'll be taking CS106A.
- You have prior programming experience at the level of CS106A or higher.

In the 2020-2021 academic year, CS105 is offered in Fall and Spring quarters.

## An Alternative Terminal Course: CS106E

CS106E is a terminal course for non-technical majors who have already taken CS106A and want to have a broader understanding of computer technology in addition to the programming the other CS106 courses focus on. The course is particularly useful to anyone that might work in the tech industry in a non-programming capacity. The course covers topics such as how the internet works, what an operating system does, how websites are built, cloud computing, artificial intelligence, computer security, and privacy.

We recommend that you take CS106E if

- You meet all the requirements for taking CS106B.
- You are fairly certain that you won't be continuing onward to a CS major in the future.

We recommend that you *not* take CS106E if

- You are thinking about majoring in CS, but would like an easier alternative to CS106B/X.

In the 2020-2021 academic year, CS106E is offered in Spring quarter.

## **I already know how to program. Shouldn't I skip the intro courses altogether?**

Many students entering Stanford today have had considerable programming experience in high school or from their own independent work with computers. If you are in that position, the idea of starting with a beginning programming course – even an intensive one like CS106B – seems like a waste of time. Your perception may in fact be correct. In our experience, there are somewhere between 10 and 15 students in each entering class who should start at a more advanced point in the sequence. Below we talk about some of these more advanced classes (CS107 and CS107E).

For most of you, however, the right place to start is with the CS106 series. Most high-school computing courses are somewhat weak and provide little background in modern software engineering techniques. By taking CS106, you will learn how the CS department at Stanford approaches programming and get a solid foundation for more advanced work. If you're unsure where you should start the programming sequence, please talk with us.

## **How it All Works: CS107**

CS107 is designed as a first course in computer systems programming. It explores how high-level programming constructs are represented internally inside the computer and how those internal representations affect program behavior and performance. Along the way, it provides programming maturity and exposure to developing software in a Unix environment.

CS107 has CS106B as a prerequisite and assumes an understanding of fundamental programming techniques and good programming style. As a result, it's rare for incoming students to jump directly into CS107 and to skip the CS106 series entirely. Typically, we'd only recommend this to students with a background comparable to CS106A/B and who already have good programming style. Most students, even those who go on to be CS majors, usually begin in CS106A/B.

We recommend that you take CS107 if

- You have completed CS106B or have the equivalent programming background, including familiarity with recursion and fundamental data structures (binary trees, dynamic arrays, linked lists, graphs, etc.)
- You have experience writing readable code – writing comments, decomposing problems into smaller pieces, etc.

We recommend that you *not* take CS107 if

- You have never before taken a class in computer programming.
- You have prior programming experience, but have not met the postconditions of CS106B.

In the 2020-2021 academic year, CS107 is offered every quarter.

## How it All Works on a Tiny Computer: CS107E

CS107E is version of CS107 that covers similar topics but which focuses on programming a small computer that can easily fit into the palm of your hand. The class is smaller and more project-oriented than CS107 and lets you play around with small embedded devices to see how low-level systems concepts directly let you control physical devices.

We recommend that you take CS107E if

- You meet all the requirements for CS107.
- You enjoy working on open-ended projects.

We recommend that you *not* take CS107E if

- You're nervous about taking CS107 and want to satisfy that requirement in a different way.

In the 2020-2021 academic year, CS107E is offered in Autumn and Winter quarters.