

CS Course Placement Information

Handout adapted by Cynthia Lee and Keith Schwarz from the original by Eric Roberts and Mehran Sahami.

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.

CS106A

CS106A is our first-quarter programming course. 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! Most students interested in learning computer programming will start here, including many eventual CS majors.

CS106A focuses on fundamental concepts in computer programming (**if** statements, methods, arrays, etc.) that arise in many different programming languages. Although CS106A is taught using the Java programming language, it is not designed as a class in standard Java programming.

In our experience, 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 or CS106X. If you're unsure about whether your background is equivalent to CS106A, you're always welcome to ask the course staff.

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 significant prior programming experience in a language other than Java, and simply want to add the Java language. (If this is the case, check out CS108.)

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

CS106B is taught using the C++ programming language, but doesn't teach many techniques specific to the C++ programming language. 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 significant prior programming experience in a language other than C++, and simply want to add the C++ language.

CS106X

CS106X is the “honors” version of CS106B. It covers the same material as CS106B at a slightly faster pace, plus some additional topics not normally covered in CS106B. In the past, CS106X used to be CS106A and CS106B combined into a single quarter, but that is not currently the case.

You shouldn't feel pressure to take CS106X if you're thinking that at some point you might go on to major in computer science. Most CS majors go through the CS106A/B sequence, with only a small number of students opting to take CS106X. Rather, we'd recommend taking CS106X only if you're really interested in the material and want to explore it in more depth.

We recommend that you take CS106X if

- You meet all criteria for taking CS106B.
- You want to get a more in-depth treatment of the topics from CS106B.
- You are willing to put in more work than is necessary for CS106B.

We recommend that you *not* take CS106X if

- You have no programming experience and are hoping to complete the equivalent of CS106A/B in a single quarter.
- You are taking it because you'd really like to take CS106B, but have a scheduling conflict that prevents you from doing so.
- You really want to take CS106B, but are concerned that you need to take CS106X to avoid fall-

ing behind everyone else (trust us, that's not going to happen.)

- You really want to take CS106B, but CS106X is in a better timeslot, or is SCPD/videotaped for convenience, or similar reasons (it really is harder, so if that's not what you're looking for you'll regret it).

CS106L

CS106L is a one-unit companion course to CS106B and CS106X 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 and CS106X, it's open to anyone with a comparable background.

We recommend that you take CS106L if

- You have prior programming experience at the level of CS106B or CS106X (or are currently enrolled in those courses.)
- You are interested in learning more about the C++ programming language.
- You're interested in seeing aspects of C++ that are not covered in CS106B or CS106X.

We recommend that you *not* take CS106L if

- You are interested in getting a deeper understanding of topics like recursion, data structures, or algorithmic analysis.
- You want to learn programming at the level of CS106B or CS106X, but don't have the time to take those courses.

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 either CS106B or CS106X as prerequisites 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/X.

We recommend that you take CS107 if

- You have completed CS106B or CS106X or have the equivalent programming background, including familiarity with recursion and fundamental data structures.
- You have good programming style.

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 never taken a college-level course in computer programming.

CS107E

CS107E is a new, experimental course that covers similar topics as CS107 but focuses more on programming the computer “from the bare metal” (that is, directly interfacing with the hardware). It has limited enrollment and the application period has already ended, so keep an eye out for this course the next time it rolls around.

CS108

CS108 is a class in object-oriented systems design. The class covers software engineering and software design techniques for working on large projects in groups. It's taught using Java and, unlike CS106A, does go into some depth about the Java programming language. At the end of the course, you'll be asked to complete a final project where you'll build a large, complex software system from scratch.

We recommend that you take CS108 if

- You have completed CS106B or CS106X or have the equivalent programming background.
- You are interested in learning how to design and test large software systems.
- You have prior programming experience and want to learn how to build large Java programs.

We recommend that you *not* take CS108 if

- You have programming experience at the level of CS106A or lower and are interested in learning standard Java programming.

CS142

CS142 is a first course in web programming. This course, which has grown in popularity each year, gives a complete picture of how web applications are built, including front-end and back-end technologies, databases, and security concerns.

Designing and building websites requires some specialized knowledge about how web browsers and web servers work, but fundamentally relies on the same programming techniques that arise in other contexts. As a result, CS142 assumes a programming maturity at the level of CS107 or CS108. Students who take CS142 without a comparable programming background often find themselves in way over their heads.

We recommend that you take CS142 if

- You are interested in learning how to design and build web applications.
- You have a programming background comparable to having completed either CS107 or CS108.

We recommend that you *not* take CS142 if

- You are currently enrolled in any of the CS106 courses or have a programming background that's only equivalent to the CS106 series.
- You do not have a strong programming background and intend to pick up the missing pieces while taking CS142. Historically, this tends to not work well at all – the CS142 staff typically doesn't provide assistance on basic programming concepts.

An Alternative: CS101 or 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) but don't necessarily want to invest a lot of time learning how to program computers, you may want to look into CS101 or CS105. These courses are designed to give a broader introduction to computing than CS106A at the expense of more significant programming depth.

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.