

Course Placement Information

Based on a handout by Eric Roberts

Computers are everywhere in today's world. The more you know about computers, the better prepared you will be to make use of them in whatever field you choose to pursue. Learning to program computers unlocks the full power of computer technology in a way that is both liberating and exciting. At the same time, programming is an intellectually challenging activity that comes easily to very few people. Taking a programming course requires a great deal of work and commitment on your part, but you will not be able to master programming without putting in that level of work somewhere along the way. The payoffs, however, are quite real. If you make the effort and keep up with the demands of the material, you will be able to make computers do amazing things. As you begin your journey in computing at Stanford, it would serve you well to decide what are the best options for you along this route.

What introductory programming course should I take?

A very large percentage of Stanford students take a programming course from the Computer Science Department at some point during their undergraduate career. Because we need to accommodate students with a wide range of backgrounds and interests, the CS department offers several different introductory classes:

- *CS 101 – Introduction to Computing Principles.* This is a new course to introduce the essential idea of Computer Science for a zero-prior-experience audience. CS101 fills a similar niche to CS105, in terms of appealing to students who take the course primarily to meet the Stanford General Education Requirement in category DB-EngrAppSci. The design of CS101 put an emphasis on creating engaging, interesting assignments that get at the key ideas of computer science by experimenting with computer code, but without bogging down too much in programming detail. This is a difficult balance to strike, but it's a priority for CS101. The CS101 assignments and exercises are geared to be done live in class. CS101 is being offered in the Spring quarter this year.
- *CS 105—Introduction to Computing.* This course is designed as a general-education introduction to what this rapidly expanding field of computer science is all about. It attracts an audience of approximately 500 students a year, most of whom take the course primarily to meet the Stanford General Education Requirement in category DB-EngrAppSci. If your only interest is in meeting that requirement, CS 105 is likely to be the most appropriate course. Like any programming course, CS 105 requires a reasonable amount of work, but not as much as CS 106A. CS 105 is offered in Autumn, and Winter quarters this year.
- *CS 106A—Programming Methodology.* This course is the largest of the introductory programming courses and is one of the largest courses at Stanford (taken by approximately 1700 students a year). CS106A teaches the widely-used Java programming language along with good software engineering principles. 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. CS106A is offered in Autumn, Winter, and Spring quarter this year.

- *CS 106B—Programming Abstractions.* This course is the natural successor to CS 106A and covers such advanced programming topics as recursion, algorithmic analysis, and data abstraction using the C++ programming language, which is similar to both C and Java. If you've taken the Computer Science AP exam and done well (scored 4 or 5) or earned a good grade in a college course, CS106B may be an appropriate course for you to start with, but often CS106X is a better choice. CS106B assumes that you already have familiarity with good programming style and software engineering issues (at the level of CS106A), and that you can use this understanding as a foundation on which to tackle new topics in programming and data abstraction. CS106B is offered in the Autumn, Winter, and Spring this year.
- *CS 106X—Programming Abstractions (Accelerated).* CS106X currently operates as an "honors" version of our CS106B course. It is taught using the C++ programming language and covers the same topics as CS106B but with more in-depth coverage in some areas. In order to get through that much material in a quarter, CS 106X moves at a very fast pace. Students are expected to have solid background comparable to our CS106A course, and should have sufficient maturity and dedication to tackle an intense challenge. If you've had previous programming experience, this class is an excellent way to learn C++ and brush up on your skills. If you haven't done much programming before or don't feel comfortable with your programming skills, you should take the CS 106A/B sequence instead. Don't let anyone tell you that "real engineers take CS 106X." These days, most computer scientists and engineers start with CS 106A, where they do just fine. The last thing you want to do is get in over your head. If you think you are ready for CS106X, you might want to see how you feel about the course expectations and pace after attending the first few classes. CS 106X is offered in Autumn and Winter quarter this year.

Note: Students who have previously taken CS106A and wish to enroll in CS106X in place of CS106B are free to do so. You just need be prepared for a fast paced course.

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 CS 106X—seems like a waste of time. Your perception may in fact be correct. In my experience, there are at somewhere between 10 and 15 students in each entering class who should start at a more advanced point in the sequence. For most of you, however, the right place to start is with the CS 106 series. Most high-school computing courses are quite weak and provide very little background in modern software engineering techniques. By taking CS 106, 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 me.

Other courses

As computers become more powerful, it is possible to use them for increasingly sophisticated tasks without engaging in programming, at least in a traditional sense. The CS 106 courses teach you about programming, and not about a particular programming language.

If your goal is knowing more about how to use computers, you should investigate the following courses:

- *CS 1C—Introduction to Computing at Stanford.* This one-unit course is offered in the Autumn quarter only and makes sure you have a level of “computer literacy” that will allow you to function effectively at Stanford. It does not teach programming at all.
- *CS 2C—Multimedia Production.* This course covers topics related to editing and publishing multimedia (sound, image, and video), including using applications like Photoshop and iMovie. Like CS 1C, this course does not teach programming and requires no programming experience.

If, on the other hand, you already have programming experience and want to learn about specific languages and tools, you should check out the following courses:

- *CS 142—Web Applications.* This course covers the concepts and techniques used in constructing interactive web applications. It requires substantial prior programming experience (prerequisites: CS107 and CS108). It will be offered during Spring quarter this year.
- *CS 193C—Client-Side Internet Technologies.* This course covers such client-side web-oriented topics as JavaScript, XHTML, CSS, XML, and Flash. The course requires previous programming experience at the level of CS106A. It is offered during the Summer quarter.
- *CS 193I—Introduction to iOS Programming.* Introduction to the design and engineering of iOS applications. Uses iOS and the Objective-C programming language. The course requires previous programming experience at the level of CS106A. It is offered during the Spring quarter this year.
- *CS 193P—iPhone and iPad Application Programming.* As the name implies, this course charts the development path for a iPhone and iPad applications. It requires prior programming experience (prerequisites: CS106B/X as well as familiarity with the C programming language). It will be offered during Winter quarter this year.