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.

What introductory programming course should I take?

A 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 range of backgrounds and interests, the department offers several different introductory classes:

- **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 250 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 the autumn and spring quarters only.

- **CS 106A—Programming Methodology.** This course is the largest of the introductory programming courses and was for several years the largest course at Stanford. CS 106A 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. CS 106A is offered every quarter; this winter, the course meets MWF at 3:15 P.M. in Hewlett 200.

- **CS 106B—Programming Abstractions.** This course covers a variety of advanced programming topics including recursion, algorithmic analysis, and object-oriented design. It uses the programming language C++, which is relatively easy to learn from a background in either C or Java. While CS 106B is designed primarily to serve as a follow-on to CS 106A, it also makes a good entry point into the sequence for students who have taken AP Computer Science A or some comparable course in high school. (If you’ve taken the Computer Science AB course and done well, you should consider skipping CS 106 altogether as described in the following section.) This quarter, CS 106B meets MWF at 11:00 A.M. in Gates B-1.

- **CS 106X—Programming Methodology and Abstractions (accelerated).** CS 106X currently operates as an “honors” version of our CS 106B course. It is taught using the C++ programming language and covers the same topics as CS 106B 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 CS 106A 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. CS 106X is offered only in autumn 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 seems like a waste of time. Your perception may in fact be correct. In my experience, there are at somewhere between 15 and 20 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. Many high-school computing courses do not cover programming as thoroughly as we do at Stanford, particularly when it comes to modern software engineering techniques and effective programming style. By taking CS 106A and 106B, you will learn how the CS department at Stanford approaches programming and get a solid foundation for more advanced work. If you’re unsure as to where you should start the programming sequence, please talk to the CS 106 course staff.

**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 in general more than about any 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 is a continuation of CS 1C and covers more advanced topics in web-page creation and applications like Photoshop, Dreamweaver, and iMovie. Like CS 1C, the continuation is a one-unit course taught in the dorms that requires no programming experience.