Course Placement Information

This handout written 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.

What introductory programming course should I take?

Almost 80 percent of Stanford students take a programming course at some point during their undergraduate career. To accommodate students with a wide range of backgrounds and interests, the CS department offers several different introductory classes:

- *CS105—Introduction to Computing.* This course is designed as a general-education introduction to computer science. It attracts an audience of approximately 300 students a year, many of whom take the course primarily to meet the Stanford distribution requirement in applied mathematical science, which satisfies GER:2b. If your only interest is in meeting that requirement, CS105 is likely to be the most appropriate course. Like any programming course, CS105 requires a reasonable amount of work, but not as much as CS106A. CS105 meets MWF 1:15 in 420-040.
- *CS106A—Programming Methodology.* This course is the largest of the introductory programming courses and, with its enrollment now over 900 a year, one of the two largest courses at Stanford. Like all the CS106 courses, this course teaches you the C programming language, which is used widely in industry and is the base from which more advanced languages such as C++ and Java have sprung. CS106A is explicitly designed to appeal to humanists and social scientists as well as hard-core techies. In fact, most CS106A 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 meets MWF at 10am in 420-040.
- *CS106B—Programming Abstractions.* This course is the natural successor to CS106A and covers more advanced programming topics such as recursion, algorithmic analysis, and data abstraction. If you're looking for a place to start the programming sequence, however, CS106B is almost certainly not the right course, even if you've taken a programming course elsewhere. CS106X is almost always a better choice. The CS106A and B courses form a logical sequence, and it is hard to jump in midstream. The second course relies heavily on the first, not only for programming concepts, but also for several tools and techniques that are specific to Stanford's approach. In particular, the course assumes that you are familiar with the libraries introduced in the CS106A textbook, *The Art and Science of C.* If you are not familiar with those libraries, you must either commit to learning that material quickly on your own or enroll in CS106X instead. CS106B meets MWF at 2:15 in TCseq 201.
- *CS106X—Programming Methodology and Abstractions (accelerated).* This course combines 90 percent of the material in CS106A and CS106B into a single course. In order to get through that much material in a quarter, CS106X moves at an incredible pace. 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, you should take the CS106A/B sequence instead. Don't let anyone tell you that "real engineers take CS106X." These days, most computer scientists and engineers start with CS106A, where they do just fine. The last thing you want to do is get in over your head. CS106X meets MWF at 11am in Gates B03.

I already know how to program—should I skip the intro courses?

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 CS106X—seems like a waste of time. Your perception may in fact be correct. In my experience, there are at least ten 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 CS106 series. Most high-school computing courses are quite weak and provide very little background in modern software engineering techniques. By taking CS106X, 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 come and talk to me in my office (Gates 192).

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

- *CS1C—Using the Macintosh.* This one-unit course is offered in the autumn quarter only and is designed to ensure that you have a level of "computer literacy" that will allow you to function effectively in Stanford's highly computerized environment. It does not teach programming at all. In the CS106 courses, we assume that you are already familiar with the Macintosh or that you can quickly come up to speed. The Macintosh environment is remarkably user-friendly, and there's not really that much you need to learn in order to use it well.
- *CS1I—Using the Internet.* Particularly with the explosion of the World-Wide Web in recent years, much of what is exciting about working with computers is related to the global information network based on the Internet. CS1I is a one-unit, winter quarter course that covers network tools and technologies from a user's perspective.

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:

- *CS193D—C++ and Object-Oriented Programming* This course covers the C++ programming language, which is based on the object-oriented programming paradigm.
- *CS193I—Internet Programming.* This course explores the core technologies behind modern Internet tools from a programmer's perspective.
- *CS193J—Programming in Java.* This course covers the Java programming language and principles of object-oriented programming.
- *CS193U—Software Engineering in C.* This course covers programming in C at a rapid pace for those who already know the basic concepts, and then follows up that beginning with a discussion of how to write programs for the Unix operating system.
- *CS193W—Microsoft Windows Programming.* This course covers the fundamentals of programming for the Microsoft Windows platform, focusing on the use of the Microsoft Foundation Class (MFC) package.