

CS106AX: General Information

This handout is based on a course information handout template used by Eric Roberts.

Instructor: Jerry CainE-mail: jerry@cs.stanford.edu

Office: Durand 319

Phone: 415-205-2242

Hours: Mondays, 3:00 – 5:00pm (in person)
Tuesdays, 2:00 – 4:00pm (online)**TA: Avi Gupta**E-mail: agupta07@stanford.edu

Office: Durand 303

Hours: TBD

Course description**CS106AX: Programming Methodologies in JavaScript and Python**

Introduction to the engineering of computer and web applications emphasizing modern software engineering principles: object-oriented design, decomposition, encapsulation, abstraction, and testing. This course targets an audience with prior programming experience, and that prior experience is leveraged so material can be covered more quickly and in greater depth.

Terms: Aut | Units: 3-5 | UG Reqs: WAY-FR | Grading: Letter or Credit/No Credit

Why Two Programming Languages?

Until 2018 or so, CS106A was taught using the Java programming language.

CS106A is generally the largest class taught on campus in any single quarter, and the class has been well liked by nearly all our students. We have a dedicated group of professors and lecturers whose full-time job is to lead the course, we have excellent undergraduate section leaders who operate as graduate student CAs, and we have a collection of well-crafted assignments to help everyone learn how to program well. Why, then would we tinker with the course at all if everything was so stable?

Computer science is one of the most dynamic disciplines in today's world. Technology changes rapidly, and it is important for our courses to remain current if they're to remain relevant. And about 15 years ago, the Java language began to show its age. When we adopted Java in the early-to-mid 2000's, it seemed likely that Java would become one of the primary languages used in the age of the World Wide Web. As things turned out, Java's model for interacting with the web—with browser-embedded programs called

applets—failed miserably. When Java failed to provide a platform for implementing web content, JavaScript came along, assumed that responsibility, and flourished. JavaScript is now one of the most popular programming languages today, and the language—at least the portion of it we’ll expose in CS106AX—is elegant and relatively easy to learn.

While there are an unbounded number of applications that require JavaScript for them to work and work well—after all, any interactive web page relies on JavaScript to implement that interactivity—there are a large class of problems for which JavaScript isn’t the best fit. Computationally intense problems—the ones that big data and machine learning platforms are designed to solve—are better suited for more general-purpose programming languages that can execute independently of a web browser.

One such language is Python, which is currently being used as the sole language of instruction in the traditional CS106A. And if it isn’t already the case, it will soon be true that most introductory programming courses at our peer institutions use Python as well.

I feel that **we can learn both**, and in the process learn about web programming, software development, and the World Wide Web to a degree that wouldn’t be very easy if we studied just one language. Virtually all modern web clients rely on JavaScript, and many web servers rely on Python to synthesize HTML pages—or at least portions of them—before delivering them to the browser that requested them. By learning both languages, we’ll be able to not only learn how to program, but to understand how modern web applications are architected and how multiple components (HTML for content, CSS for formatting, JavaScript for interactivity, and Python for content generation) are married so that web properties like Facebook, Instagram, Google, and Amazon all work the way they do.

Lectures

Lectures are scheduled for Monday, Wednesday, and Friday at 1:30 P.M. in Mitchell B67. Lectures aren’t being recorded this quarter, since the course is small and still somewhat experimental. The lecture schedule for the entire quarter is provided via another handout.

Discussion Sections

In addition to lecture, you must also sign up for a weekly 50-minute section, which will be scheduled to begin in the second week of the quarter. A [section signup link](#) will go active this coming Thursday at 5:00P.M. and stay active until Sunday at 5:00P.M. During that 72-hour window, you can see what section times are available and list your preferences. Note there is no advantage to submitting preferences before others, so you’re welcome to wait until Sunday afternoon if need be. Please do sign up for a section before the Sunday, 5:00P.M. deadline, else you’ll be excluded from the initial round of section assignments. (Note that you should not sign up for discussion sections on Axxess, as we manage discussion sections internally.)

Units

If you are an undergraduate, you are required to take CS106AX for 5 units of credit. If you are a graduate student, you may enroll in CS106AX for 3 units if it is necessary to reduce the unit count for administrative reasons. Taking the course for reduced units does not imply any change in the course requirements. CS106AX can be used to satisfy any requirement that CS106A does.

Drop/add deadlines

You may not add or drop courses from your study list after 5:00P.M. on Friday, October 13th, without having that course appear on your transcript with a notation indicating that you have withdrawn. The last day to change your status to CR/NC or to withdraw from the course is Friday, November 17th at 5:00P.M. A more complete calendar of relevant dates can be found [right here](#).

Web page

The web page for CS106AX is, unsurprisingly, <http://cs106ax.stanford.edu>. All the materials and course announcements will be posted on this website, so be sure to visit it often.

Course materials

The required text for this class is a relatively new book called *Introduction to JavaScript Programming: The “Nothing but a Browser” Approach*, which is available at the Stanford Bookstore. The book is by Eric Roberts, who recently retired from Stanford after nearly 30 years. He spent the last several quarters of his time at Stanford working on this textbook, and it’s beautifully written.

We’ll also assign reading from a second textbook by Eric Roberts, one he worked on immediately following his retirement. The book is in draft form, and much of it is a cannibalization of his JavaScript textbook. Its treatment of the Python programming language, however, is so clear and entertaining that I want to use it anyway. The course reader PDF is available free of charge to all CS106AX students, and I’ve included a link to it on the course website.

Programming assignments and problem sets

This quarter, CS106AX requires eight programming assignments, which are due on the dates given in the calendar handout. Except for the last assignment (which is due at the very end of the quarter), each assignment is graded by your section leader, who assigns two grades—one for functionality and one for style—drawn from this list:

- ++ An absolutely fantastic submission of the sort that will only come along a few times during the quarter. To ensure that this score is given only rarely, any grade of ++ must be approved by the instructor and TA.
- + A submission that exceeds our expectations for the assignment. The program must reflect additional work beyond the requirements or get the job done in a particularly elegant way.
- √+ A submission that satisfies all the requirements for the assignment.
- √ A submission that meets the requirements for the assignment, with a few small problems.
- √– A submission that has problems serious enough to fall short of the requirements for the assignment.
- A submission that has extremely serious problems, but nonetheless shows some effort and understanding.
- – A submission that shows little effort and does not represent passing work.

From past experience, we expect most grades to be √+ and √. Dividing the grades into categories means that your section leader can spend more time talking about what you need to learn from the assignment and not have to worry about justifying each point.

For some assignment, you will need to make an appointment with your section leader for an interactive grading session. Your section leader will explain how to schedule these sessions and go over the grading process in more detail.

Examinations

The midterm examination will be administered at two different times on **Wednesday, November 1st from 3:45 to 5:45 P.M.** and from **7:00 to 9:00 P.M.** The final examination is tentatively scheduled to be offered on **Monday, December 11th, from 8:30 to 11:30 A.M.** The examinations are designed so that you should be able to complete them in less than the fully allotted time: the midterm is designed to take 75 minutes, and the final is designed to take 100. Both exams are closed computer but open book, and you may use any notes or materials from the class.

Late policy

Assignments are submitted electronically as described in the first assignment handout, which goes out this coming Monday. **All assignments are due at 5:00 P.M. on the date indicated on the assignment handout. Assignments submitted after 5:00 P.M. will be considered late.**

Because each of you will come upon some moment during the quarter where so much work piles up that you need a little extra time, we grant each of you three free "late days". To avoid any ambiguity, a "day" is defined as a class day. Thus, if your assignment was

due on Monday but turned in the following Wednesday, that assignment would be one day late. After your late days for the quarter are exhausted, programs are assessed a late penalty of one category point per late day used (a $\sqrt{+}$ turns into a $\sqrt{}$, and so forth). Late days are valuable, and it pays to save some for the more demanding assignments toward the end of the quarter. In all cases, assignments must be turned in within a calendar week of their published due date.

In special circumstances such as extended medical problems or other emergencies, extensions may be granted beyond the late days. To request an extension, send an email to agupta07@stanford.edu no later than 24 hours before the program is due. Please don't ask your section leader, as Avi is the only one authorized the grant extensions.

Grading

The most important components of the final grade in CS106AX are always the programming assignments, which count for well over half the final grade. Even so, one of the complaints we regularly hear from students is that the assignments don't count for enough relative to the exams. Many students feel that if they can implement the assignments correctly, it shouldn't be necessary to suffer through an examination taken under arbitrary time constraints without the aid of a computer.

Although we're sympathetic, there is a big problem. Computer science courses—here at Stanford as well as at most other colleges and universities—have been marked by an intolerably high number of academic dishonesty cases. Given that reality, we use exams to ensure our students have really learned the material. Someone who copies their assignments from someone else may do very well on those assignments (assuming we don't catch them) but will in all probability do poorly on the midterm and the final.

Here, then, are the weights for the different components of the course:

Programming assignments	65%
Final examination	20%
Midterm examination	10%
Section attendance and participation	5%

Fair Access

Students who need academic accommodations based on the impact of a disability should initiate a request with the Office of Accessible Education. Professional staff will evaluate the request with required documentation, recommend reasonable accommodations, and prepare an Accommodation Letter dated in the current quarter. Students should contact the OAE as soon as possible since timely notice is needed to coordinate accommodations. The OAE has contact information on their web page: <http://oae.stanford.edu>.

Financial Aid

All students should retain receipts for books and other course-related expenses, as these may be qualified educational expenses for tax purposes. If you are an undergraduate receiving financial aid, you may be eligible for additional financial aid for required books and course materials if these expenses exceed the aid amount in your award letter. For more information, review your award letter or visit the Student Budget website.