

CS 106AJ — General Information

This handout is based on a handout written by Eric Roberts in early April, 2018.

Professor: Jerry Cain

E-mail: jerry@cs.stanford.edu
Office: Gates 192
Phone: 415-205-2242
Hours: Wednesdays, 3:45 – 5:45 P.M.

TA: Ryan Eberhardt

E-mail: rebs@cs.stanford.edu
Office: TBD
Hours: TBD

Course description

CS 106AJ: Programming Methodology in JavaScript

Introduction to the engineering of computer applications emphasizing modern software engineering principles: object-oriented design, decomposition, encapsulation, abstraction, and testing. Emphasis is on good programming style. This course covers the same material as CS 106A but does so using JavaScript, the most common language for implementing interactive web pages, instead of Java. No prior programming experience required. Enrollment limited to 85.

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

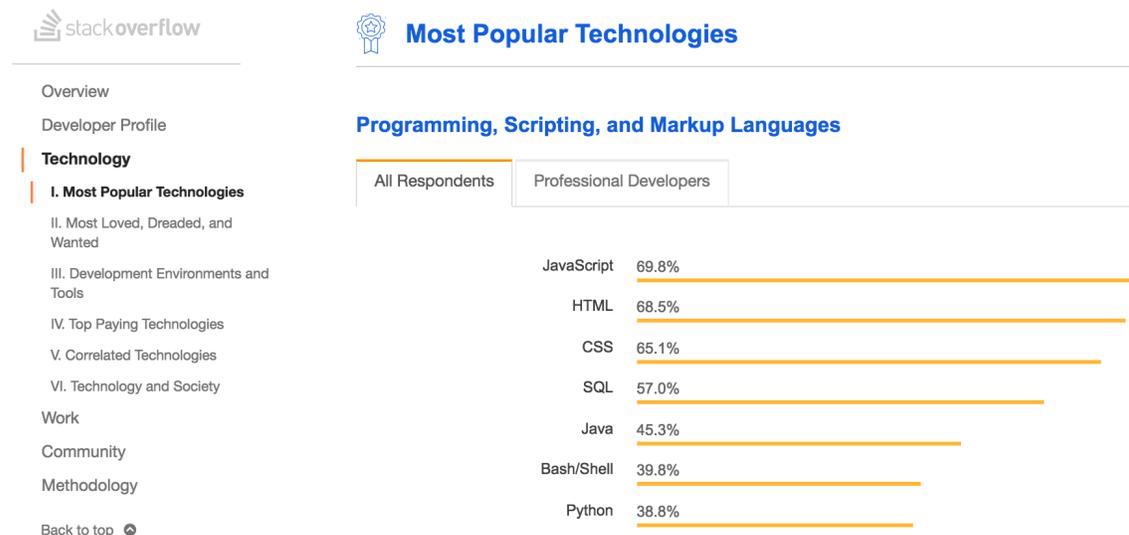
Why JavaScript?

For several years, CS 106A has been one of the largest courses at Stanford as well as one of the university's most popular. We have great faculty and lecturers teaching the classes, and virtually every undergraduate at Stanford takes at least one course from our department. The current version of the class is working very well. So why would we think about changing it?

Computer science is one of the most dynamic disciplines in today's world. Technology changes rapidly, and it is important for our courses to avoid sinking into obsolescence. The Java language—which has been the programming language used in CS 106A for the last decade—is beginning 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 through the use of programs called applets did not succeed. When Java failed to provide a platform for implementing web content, the language that rose to take over that responsibility was JavaScript, which has become the most widely used language in industry today, as shown in Figure 1.

Our reason for experimenting with JavaScript in CS 106AJ, however, goes beyond its success in the web domain. At its core, JavaScript is a highly expressive and flexible programming language with a surprising level of elegance and beauty. We believe that it will be substantially easier to learn than Java, mostly because there are fewer extraneous concepts that get in the way of understanding the fundamental concepts of programming.

Figure 1. Most popular programming languages according to stack overflow



Source: <https://insights.stackoverflow.com/survey/2018/#most-popular-technologies>

Our goal this quarter is to repeat and improve upon two prior experiments—experiments that went exceptionally well—and confirm that JavaScript works well as a first programming language.

Lectures

Lectures are scheduled for Monday, Wednesday, and Friday at 2:30 P.M. in the School of Education, Room 128. Lectures will not be recorded. The schedule for individual lectures is given on the accompanying calendar handout.

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 5:00 P.M. and stay active until Sunday at 5:00 P.M. During that 72-hour window, you can see what section times are available and state your preferences. Note that there is no advantage to stating your preferences before others, so you're welcome to wait until Sunday after to express those preferences if need be. Please do sign up for a section before the Sunday, 5:00 P.M. deadline, else you'll be excluded from the initial round of section assignments.

Units

If you are an undergraduate, you are required to take CS 106AJ for 5 units of credit. If you are a graduate student, you may enroll in CS 106AJ for 3 units if it is necessary for you to reduce the unit count for administrative reasons. Taking the course for reduced units does not imply any change in the course requirements.

Drop/add deadlines

You may not add or drop courses from your study list after 5:00 P.M. on Friday, October 12, without having that course appear on your transcript with a notation indicating that you have withdrawn from the course. The last day to change your status to CR/NC or to withdraw from the course is Friday, November 16 at 5:00 P.M.

Web page

The web page for CS 106AJ is, not surprisingly, <http://cs106aj.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 near-final draft of a new book, tentatively titled *Understanding Programming through JavaScript*, which is available from the Stanford Bookstore. Note that last year's course reader—the one used the last time CS106AJ was taught—is now obsolete, so you'll need to purchase a new reader that's only available at the Bookstore.

In addition to the textbook, we will also distribute additional material in the form of class handouts. After class, any extra copies of the handouts will be placed in the "Handout Hangout" bins in the entryway to the Gates B-wing between Gates 182 and 188. The handouts are also available in PDF format on the CS 106AJ web site. If you miss a handout in class, you can print your own copy from the web.

Examinations

The midterm examination will be administered at two different times on **Tuesday, October 30th from 3:30 to 5:30 P.M.** and from **7:00 to 9:00 P.M.** The final examination will be offered on **Monday, December 10th, 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 one hour, and the final is designed to take 100 minutes. Both exams are closed-computer, open-book, and you may use any notes or materials from the class.

Programming assignments and problem sets

This quarter, CS 106AJ requires six programming assignments, which are due on the dates given in the calendar handout. Except for Assignment #6 (which is due at the very end of the quarter), each assignment is graded during an interactive, one-on-one session with 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 job well done.
- √ 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 each assignment, you must 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.

Late policy

Assignments are submitted electronically as described in the first assignment handout.

Important note: All assignments are due at 5:00 P.M. on the date indicated on the assignment handout. Assignments submitted after 5:00 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, every student begins the quarter with two free "late days." To avoid any ambiguity, a "day" is defined as a class day. Thus, if your assignment was due on Friday but turned in the following Monday, 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 √+ turns into a √, and so forth). Late days are valuable, and it pays to keep some around for the harder 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 rebs@stanford.edu no later than 24 hours before the program is due. Only Ryan is authorized to approve extensions. In particular, do not ask your section leader.

Important note: You may not use any late days—free or not—for the last assignment, which is due on the last day of class.

Contests

As shown on the calendar handout, there are three contests scheduled at different points during the term. The point of these contests is to give you a chance to show creativity and initiative beyond what is formally required by the course. Rules for each contest will be distributed in class when they are announced.

To encourage greater participation in the contests, we will offer two additional incentives. First, every serious entry gets you one virtual ticket in a random drawing for a special grand prize at the end of the quarter. The more contests you enter, the more chances you have. Winning runner-up prizes or honorable mentions in a contest or submitting

assignments nominated for ++ scores give you additional chances. The random drawing will take place at the beginning of the review section for the final exam.

In this experimental version of CS 106, you have an additional opportunity to win virtual tickets. The draft reader has not yet been fully edited and will certainly contain errors. Similarly, the programming environments are new and will certainly have bugs. If you find a bug in the reader, please send it to both eroberts@cs.stanford.edu and jerry@cs.stanford.edu. If you're the first person to report a genuine problem, you get a ticket for the random drawing.

Grading

The most important component of the final grade in CS 106AJ is always the programming assignments, which count for just 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 as long as 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 with this argument in theory, there is a 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 as a check to ensure that students have learned the material. Someone who copies their assignments from someone else may do very well on those assignments (assuming we don't catch it) 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	55%
Final examination	25%
Midterm examination	15%
Section participation	5%