CS193D Course Information

Instructor: Jerry Cain
Email: jerry@cs.stanford.edu
Office phone: 650-725-8597
Home phone: 415-333-4277 (after 7 a.m. and before 10 p.m.)
Office: Gates 189
Office hours: Tuesdays, 1:00 - 2:30 p.m.
Thursdays, 1:00 - 2:30 p.m.

Don't take the above hours so seriously. If you're hankering to see me and you want to know if I'm around, just dash off an email or call me to see if I'm busy. I'm generally happy to help you as long as I don't have a pressing deadline to meet.

Lectures: TTh 11:00 - 12:15 p.m. (live broadcast on SITN channel E4)
Gates B01

Sections: There is an optional review section held on Thursdays at 4:15 p.m. in Gates B01. The review section will present examples and take time to work through student questions. The section is optional but recommended. The section will also be televised, so SITN student can also partake in the fun. Just tune in to Channel 2.

Prereqs: The prerequisite for the class is programming and problem solving at the CS106B/X level. I'm assuming you are proficient in C—in particular, you have an excellent understanding of arrays, strings, pointers, and dynamic memory allocation. You should be comfortable with recursion, you should be able to construct and manipulate linked data structures (lists, queues, trees, graphs, etc.), you should be familiar with abstract data types and you should have an understanding of various algorithmic techniques such as searching, sorting, and hashing. You will be expected to produce elegant, well-decomposed, commented solutions as taught in the CS106's.

Readings: No textbooks are required, but you may want to pick up a copy of any one of the following textbooks for reference. My assessment of the lot of C++ textbooks is that none of them are really very good, since they either assume no prior programming experience whatsooever, or they assume you've been programming in C++ for the past 10 years. There will be

times, however, when you'll just want to have a book handy to look up prototypes, read up on something that wasn't clear in lecture, etc. Here's a list of textbooks I recommend:

1. Deitel & Deitel, C++: How to Program,Prentice Hall, 2000.
2. Satir, Gregory and Brown, Doug, C++, The Core Language, O'Reilly, 1996.
3. Oualline, Steve, Practical C++ Programming, O'Reilly, 1997.
4. Eckel, Bruce, Thinking in C++, Prentice Hall, 2000, 2nd edition.
5. Stroustrup, Bjarne, The C++ Programming Language, Addison-Wesley, 1994.
6. Lippman, Stanley B., C++ Primer, Addison-Wesley, 1998, 3rd edition.
7. Austern, Matthew H., Generic Programming and the STL, Addison-Wesley, 1999. (covers only the STL very well, but only coveres the STL)

The Deitel & Deitel text just came out recently, and while I haven't read the book in full, I have read enough to see that past CS106 students will enjoy this text if they enjoyed Eric Roberts' text, since they adopt extremely similiar methodologies and the examples, while not quite as strong, are still very clear and helpful. If you way more to read, then Eckel text is pretty good, and then Lippman text is always clear and easy to read. The Austern text covers a small but incredibly fascinating subset of C++ that the first six texts hardly cover, so I recommend you pick a copy of that up after the quarter is over if you intend to do some serious C++ programming.

All these texts are available at http://www.amazon.com, and can be shipped directly to your dorm room within 48 hours.

Web site: Check out the one-stop-shopping CS193D web site at http://cse.stanford.edu/class/cs193d/. Using any Web browser, you can grab copies of handouts (published in Adobe PDF), download the assignment files, check out the current syllabus, and get general course information. We will also use the assignment area of our web site to post general-interest question/answers, clarifications, hints, etc. about assignments. Visit our site and tell us what you think! Woo!

Software: The projects for the class should be portable in that the C++ code required should work on any contemporary C++ compiler. We will concentrate our efforts on supporting **gcc 2.95.2** on Unix. You are welcome to use other compilers or computers, but bear in mind that compilers differ from machine to machine and C++ compilers are still evolving. If you are using an unsupported platform and run into

difficulties, be prepared to switch to the one I'm supportiong. No matter which platform you use, you are responsible for ensuring your final program will compile and run on the **leland** workstations, since that is where all assignments will be submitted and tested.

email: If you do not already have an account on a mainframe computer from which you can send email, open an account on **leland**. Handouts available at Sweet Hall and Tresidder computer clusters explain the procedure used to open an email account. SITN students may use a Stanford email account or any email account provided by your company.

Class email: Sometime very soon, send mail to majordomo@lists.stanford.edu, and place the text **`subscribe cs193d`** in the body of the message. Doing so will add the sending email account to the CS193D mailing list. When an important announcement just can't wait until the next lecture, I'll often broadcast the message in email form to everyone who is on this list. Note that only Jerry and the TAs can post to this account. If you have general questions regarding the class, you should use either the class newsgroup to talk with students, or cs193d@cs to interrogate the staff. (More on that below!)

Staff email: We'll maintain a universal email question queue at cs193d@cs.stanford.edu. This is the appropriate address to send questions, clarification requests, comments, flames, etc. to the course staff. Answers to questions of general interest will be added to the current assignment information on the web page for the edification of others.

Newsgroup: Our class newsgroup is **su.class.cs193d**. This is an unmoderated newsgroup and students can use it to discuss issues related to the course amongst themselves. Note that the staff will rarely read or post to the newsgroup. We will answer questions sent directly to cs193d@cs and post answers to frequently asked questions on the web site.

Grading: This class is offered with both the graded or CR/NC option. The course grading is divided between programming assignments, one midterm, and a final exam. The approximate grade breakdown is:

Programs	40%
Midterm Exam	20%
Final	40%

To receive a passing grade, you must complete passing work on the final exam. In particular, if you fail the final, you will fail the course, regardless of how well you perform on the assignments and the midterms.

Exams: The midterm will be held on the evening of **Thursday, November 2nd** at 7:00 p.m. The exams will be **closed**-book, **closed**-note and is being given out of class with the idea that you may take as much time as you need to complete an exam written to be taken in 60 minutes. SITN students residing in the Bay Area are expected to take the exam here at Stanford, but remote students may take the exam on site. Alternate exams will be given earlier in the day for anyone who has a legitimate conflict with this out-of-class midterm. If you need to take the exam at another time, simply email me at least 48 hours in advance, and I'll do my darndest to accommodate your situation.

The final will be Thursday, December 14th from 3:30 – 6:30 p.m.—the regularly scheduled spot! All students will be expected to take the final at the official time, as there will be no alternate exam, so please plan accordingly. The final will be also be a **closed**-book / **closed**-note examination. SITN students residing in the Bay Area should come to campus to take the exam, but remote SITN students may take the final wherever they choose.

Exam locations will be announced as their time approaches.

Late Policy: The class material builds on itself and getting behind can be hard to recover from since it tends to propagate through the following assignments by a domino effect. As a result, late work is highly discouraged— any assignment turned in late will be assessed a penalty of 5% per day (24-hour period). Assignments will not be accepted more than two days after the original assignment due date.

That being said, there are unforeseen emergencies (illness, bike accidents, disk crashes, network troubles, etc.) that cannot always be planned for in

advance. Instead of having to ask for special allowances on an individual basis, I will give each of you the privilege of granting yourself small extensions whenever crises arise. You will have two "late days" (24-hour periods) which you may use to extend the due dates of any assignment without penalty. You may use one or even both late days for any particular assignment, or you may split them up and use them for two assignments.

Honor Code: Although you are encouraged to discuss ideas with others, your programs are to be completed independently and should be original work. Whenever you obtain significant help (from other students, the TAs, students in other classes) you should acknowledge this in your program write-up, e.g. "The idea to use an insertion sort to alphabetize the names came from a discussion with my awesome TA, Eric Rager Ranelletti." Any assistance that is not given proper citation will be considered a violation of the Stanford Honor Code. The Honor Code is taken very seriously in this class; any problems will be swiftly referred to the Office of Judicial Affairs.

To be even more specific**,** you are not allowed to collaborate on the coding of your programs, nor are you allowed to copy programs or parts of programs from other students. The following three activities are among what I consider to be Honor Code violations in this course:

1. Looking at another student's code.
2. Showing another student your code.
3. Discussing assignments in such detail that you duplicate a portion of someone else's code in your own program.