

CS193k — Advanced Java

<http://www.stanford.edu/class/cs193k/>

The Course in a Nutshell

CS193k selects a few interesting areas of advanced Java technology and explores them in moderate depth. CS193k is not a basic Java course, and we will not review Java or OOP basics (see Prerequisites below). Instead, the course covers advanced Java topics which basic Java courses never get to. For each topic area, we will cover how to use it, what it's good for, related implementation and language issues, and then we'll have a non-trivial assignment.

Topics

- JFC/Swing. Use Java to build portable, dynamic, library based GUIs. Concepts include the paint() cycle, layout managers, listeners, Model-View-Control structures, JFC threading. Related advanced topics include serialized GUIs, beans, and reflection — these will probably get delayed until the end of the term.
- Threading. Work through some heavy mutual exclusion and synchronization problems. Examine the deprecation of stop(). Threading is harder to get right than most Java programmers realize.
- Distributed Computing / RMI -- build distributed computing applications on RMI -- 1-1 and n-1 Client/Server.
- Miscellaneous Advanced Topics -- XML in Java, VM implementation issues, Collections, performance techniques, beans, reflection, serialization. These topics will be covered during the quarter as needed, or in the "Miscellaneous" lectures at the end of the term.
- Things we probably won't cover in any depth -- Java 2d, JNI, Java 3d, Servlets (covered in CS193i).

Preparation and Prerequisites

The prerequisite for the course is a reasonable understanding of basic Java and OOP design. Taking CS193j or CS108 is fine preparation. Alternately, people with generally strong programming backgrounds (CS107), may take the course so long as they pick up the Java basics on their own -- for example, working through the Java tutorials below.

Java Documentation

There are many books and online resources for Java. No book is required for CS193k since there's so much available online. Here are a few resources to get you started...

<http://java.sun.com/docs/books/tutorial/index.html> -- the "Java Tutorial" of language basics. The "Language Basics" and "Essential Classes" tracks represent things you should know before taking 193k.

<http://www.afu.com/javafaq.html> -- Peter van der Linden's comp.lang.java FAQ -- lots of useful tidbits.

<http://mindprod.com/gloss.html> -- Roedy Green's Java Glossary also with lots of useful tidbits. Please note that CMP now stands for Canadian Mind Products and not Consolidated Moose Pasture.

book: Mastering Java 1.2, by John Zukowski. Very complete coverage of Java 1.2. 1200 pages and \$50. One of the thickest, heaviest books I own. If you don't like reading things online, buy this book since it includes documentation for practically everything in CS193k. You'll still need to read some things online of course.

book: Just Java 1.2, by Peter van der Linden. A thinner book with slightly less complete coverage -- it's missing RMI.

<http://cslibrary.stanford.edu/104/> -- the world famous Binky Pointer Fun video.

Platform

We will aim for the Java 1.2 standard (renamed to be "Java 2" in some sense that nobody understands), but actually a 1.1 VM can be made to work for most purposes. You can write the code on the platform of your choice, but we will test it against the Solaris VM on the elaines, so that's what you are ultimately responsible for. If for some strange reason you do not wish to be physically near an elaine for your final testing, you can still use VNC to remote-host on an elaine -- see the VNC install in /usr/class/cs108. VNC may get installed somewhere more official in the leland file system if the maintainers should give me permission to. As practical matter, the areas of Java we explore are quite portable (i.e. not AWT), so portability should not be too much of a problem. You will need a leland account to do your submissions — call (650) 725-2101.

Homeworks

There will only be 3 or 4 homeworks -- basically one for each technology area. This is a 2-unit course; it will be much more work than a typical 1-unit seminar. However, it should be less work than a typical 3-unit CS course. The homeworks themselves will be non-trivial, but there will not be too many of them.

Online Materials

In keeping with our shiny new electronic information planet, all class materials (handouts, examples, homeworks...) will be available from the course web page at <http://www.stanford.edu/class/cs193k/>, and all submissions will be electronic. I generally give out an outline for each lecture which sketches out what I have in mind for that day. If I'm going to present a lot of code, then I'll have the full listing in the handout. I will try to put the electronic copies of each day's handouts up on the web by 11:30 a.m. before lecture -- remote students may want to bring them up on screen or print them before lecture in order to follow along. Not that my handwriting is that bad. Well ok, it is.

Grading

The final course grade will be computed from approximately 50% homeworks, 50% final exam. You must pass both the homeworks and the final to pass the course. There will not be a midterm. Remote students may take the final exam remotely. Our final exam will be in its regularly scheduled slot. There will be an alternate sitting of the exam immediately preceding its regular slot.

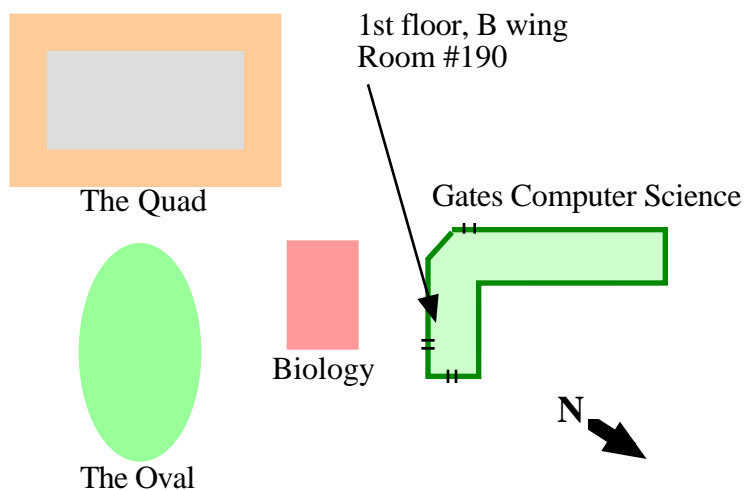
Lecture

We meet once per week from 12:30-2:05 in Skilling Auditorium and on channel E2.

Instructor

Nick Parlante
nick.parlante@cs.stanford.edu
<http://www-cs-faculty.stanford.edu/~nick/>
 (650) 725-4727

Nick's Office: Gates 190. On the first floor, facing the green Biology building.



I'll list my regular office hours along with the staff's on a separate handout. However, I'm often in my office in afternoon and evening — feel free to call or stop by.

Paper Handouts

I will provide ample paper copies of the handouts for all who attend class in person + 20% or so. Leftover paper copies of the handouts from class are kept in the bins down the hall from my office. Once those run out, please use the electronic versions— I am not committing to keeping the paper bins perpetually full. I'll make plenty for class time, and when they're gone they're gone. On the other hand, there's no handout fee.

Email Question Address

We'll maintain a universal e-mail question queue at `cs193k@cs.stanford.edu`. Answers to questions of general interest will get CC'd to the class newsgroup. If your question is going to require stepping through code, looking at variables, etc...please bring it to office hours so someone can look at it properly. When framing your question, try to articulate what you are trying to do, what you have tried, and what you think is going wrong. Short, specific questions work well by email. More involved questions work best by coming to office hours, or calling during office hours so at least there's a dialog. I will provide a handout summarizing the time, location, and phone number for all of the staff hours once we get that sorted out.

Newsgroup

We have a class newsgroup at `su.class.cs193k`. In order to keep the newsgroup low traffic/high content for such a large class, students should not post questions there. Send questions, comments, etc. to `cs193k@cs.stanford.edu`. We will post abstracts of the most interesting questions and other announcements to the newsgroup. Think of the newsgroup as read-only for the students and write-only for the staff.

Late Submissions

Instead of having to ask for extensions on a catastrophe by catastrophe basis, everyone gets three calendar “late days” to extend the due dates of any of the assignments. In keeping with the all electronic, 24-hours a day theme of the post-Internet world, late days will be measured in straight calendar days with no distinction for weekends or holidays. All homework deadlines will be at midnight Pacific time. (The semantic nit in the audience will note that due to the start of daylight savings time, at some point in the quarter you will *lose an hour*. Any student concerned about this can bring a Federation approved temporal containment module to my office at the end of the quarter, and I will refund the hour.)

These late days are intended to deal with the ordinary events of student life, both frivolous and serious: 2 midterms that day, inadvertently spent all night playing WarCraft II, disk crash, med. school interview, illness, started way too late...After your late days are used up, late work loses pretty quickly— about a half a letter grade per day. Come and see me in person in exceptional circumstances. Note that disk failure, accidental `rm*`, and other computer or network problems probably *do not* represent exceptional occurrences. Hoard your late days “just in case”, or spend them early and fly with no parachute— it's up to you.

Giving students their own late-day supply seems more fair since all the students are on the same footing. However it means you now need to make your own decisions about when to use a late day, and when to just turn in what you have. It should allow you to do a better job and hopefully learn more in the cases where your schedule gets disrupted. However, three late days do not provide too large a cushion. You should plan to finish your homeworks on time and reserve the late-days for real problems.

By default, I'm assuming that SITN students and all other non in-class-in-person-the-traditional-way students have exactly the same deadlines as everyone else. The handouts and materials go up on the web at the same time planet wide, and typically more than a week before the assignment is due. TVI or other SITN students with exceptional latency problems should contact their TA (once they've been assigned) to work out a schedule to account for their logistical delay.

Honor Code

You are free to discuss ideas and problem approaches with others, but all the work you hand in should be your own creation. **In particular, sharing or copying code is not OK.** If you feel a particular bit of collaboration may have crossed the line, just clearly cite what help you got and from whom in your submission README. You can never get in Honor Code trouble if the help is clearly credited.

Class Quotes

"People tell me one thing one day and out the other."
znu

"C++ — the power elegance and simplicity of a hand grenade"
Kenneth Dyke

"Beware the lollipop of mediocrity. Lick it once and you suck forever."
Rich Siegel

Lecture Plan

Here's the topic plan for CS193k -- the schedule may shift by a week one way or the other depending on how quickly we proceed. The assignment for a topic will generally be due on the Thursday a week after the end of the discussion of that topic. Assignments will go out one or two weeks before they are due. To make up for the many, many times I have taught a course where there was a large project due in week 10, CS193k will not have an assignment due week 10.

Week / Tue	Topics
1 Mar 28	Introduction -- what the course is about, tour some tricky areas of Java, Swing background
2 Apr 4	Swing 1 -- paint() cycle, layout managers, basic Swing components
3 Apr 11	Swing 2 -- event/listener system, MVC structure
4 Apr 18	Threading 1 -- tricky mutual exclusion and synchronization, locks, green vs. native threading models (Swing due)
5 Apr 25	Threading 2 -- wait() and signal(), building a counting semaphore, deprecation of stop(), using interrupt()
6 May 2	Threading 3 -- finish threading
7 May 9	RMI 1 -- basic client server structure through RMI, threading, serialization, and interruption issues. (Threading due)
8 May 16	RMI 2 -- finish RMI
9 May 23	Advanced 1 -- survey other Java technologies, VM implementation issues, Java dynamic Collections vs. traditional compile-time typed approaches, introspection, Java inertia and politics, performance issues. (RMI due)
10 May 30	Advanced 2 -- continue topics above.
Finals	Final exam: Sat June 3, 3:30-6:30 pm There will be a single alternate time for the exam, but it will be near to the regular time, so you will need to be in town. The alternate will probably be late the night before the regular time. You must be able to attend one of the two times to take the exam.