CS193k, Stanford
Spring, 2000-2001

Handout #1
Nick Parlante

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 basic Java or OOP (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

- Building GUIs with JFC/Swing. We'll have a quick review of the basics of layouts, drawing, and controls, and then concentrate on the Model-View-Controller paradigm in the Swing library.

- Concurrency/Threading. Work through "classical" mutual exclusion, synchronization, and interruption in Java. Also examine the more modern issues of threading a GUI. 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.

- For Servlets and JSPs, take CS193i.

- Miscellaneous Advanced Topics -- there is time for a few miscellaneous topics at the end of the quarter. Here are some possible topics: VM implementation issues, performance techniques, reflection, Javadoc, Java Web Start, features coming in Java 1.4.

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. There is no formal required book for CS193k since so much of what we need is available online, though there are many good books available for those who want them...

book: Core Java Vol II, Advanced Features by Cay Horstmann. Includes at least a chapter for most of the cs193k topics.

http://www.bruceeckel.com/TIJ2/ -- the complete text of Bruce Eckel's Java book, free in online form.

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.

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

Platform

We will code for the JDK 1.2 VM, but actually a 1.1 or 1.3 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 1.2 VM on the elaines, so that's what you are ultimately responsible for. The path to the 1.2 VM on the elaines is "/usr/pubsw/apps/jdk-1.2.2/bin " -- add that to your path in your .cshrc. Here's what that part of my .cshrc looks like...

```
set path=( \
    /usr/pubsw/apps/jdk-1.2.2/bin \
    /usr/class/cs108/bin \
    $site_path \
    . \
)
```

Use "which java" to verify which version of the JDK you are using. 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/bin. 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 be 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. The course page will include all manner of information of interest to CS193k, including assorted Java links and FAQs for the assignments. I generally give out an outline handout 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 at least an hour 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 40% homeworks, 60% 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 (see the schedule on the last page).

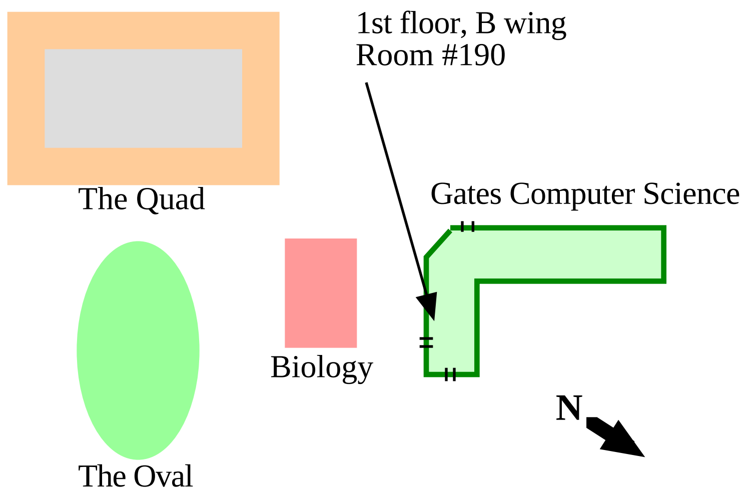
Lecture

We meet Tue 12:30-2:05 in Terman Auditorium. Please sit near the front, as the room is absurdly too large.

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`. If a question is common enough, we'll add it to the FAQ list on the course page. 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 be 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.

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.

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

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

"Somewhere between 'Live Free Or Die' and 'Famous Potatoes', the truth lies."

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. In this initial schedule, the assignments are spaced every 2 weeks.

Week / Tue	Topics
1 Apr 3	Introduction -- what the course is about, tour some tricky areas of Java, Swing background
2 Apr 10	Swing 1 -- quick coverage of basic Swing: layout managers, drawing, the repaint cycle, controls and listeners, start Model-View-Controller
3 Apr 17	Swing 2 -- more Model-View-Controller structure: JTable. Start threading
4 Apr 24	Threading 1 -- mutual exclusion and synchronization, locks, wait, notify (Swing due)
5 May 1	Threading 2 -- building a counting semaphore, deprecation of stop(), using interrupt(), threading in Swing
6 May 8	RMI 1 -- basic client server structure through RMI, threading, serialization, and interruption issues. (Threading due)
7 May 15	RMI 2 -- finish RMI
8 May 22	Advanced 1 -- pick and choose among miscellaneous advanced topics such as the following: VM implementation issues, what's coming in Java1.4, introspection, Java inertia and politics, performance issues and tuning, Java Web Start. (RMI due)
9 May 29	Advanced 2
10 Jun 5	Advanced 3 (Misc Advanced due)
Finals	**Final exam: Tue June 12th, 3:30 pm**