

CSC207.01 2014S, Class 56: Review for Final

Overview

- Preliminaries.
 - Admin.
 - Questions.
- Some questions on the course.
- Some things you didn't learn.
- Preparing for the final.

Preliminaries

Admin

- Food (donuts and grapes are leftovers)
- Notes on both finals appear in yesterday's eboard.
- Congratulations to our almost-graduating seniors.

Upcoming Work

- Fill in the final preference sheet by Monday.

Extra Credit

- CS table Friday: Casual conversation.
- Conference track meet Friday and Saturday. NBB runs at 4:05 and 5:10.
- Listen to EB's radio show on KDIC Friday at 5pm.
- Listen to DNP guest star on some radio show Friday at 11pm.
- Saturday at 2pm: Make Recycled Notebooks, food from Casa Garden or Choung Margaritas (but no alcohol).

Some questions on the course

- What's something I did well?
 - Eboards.
 - Web site. We didn't have to talk to you.
 - Very detailed on exams and resolved any ambiguities.
 - Stupid student and professor names on tests.
 - Sam has as bad work habits as students and therefore answers email at 11pm and 5am.
 - JSON project allowed creativity.

- What else should I improve?
 - Need to more frequently call people who don't show up.
 - Tests didn't reflect enough what we did in class or in the readings. Too much iterators.
 - End of semester: Project, Exam, Final. Too back-loaded. Maybe exams week 6 and 10. Project weeks 11-13. Final. Maybe optional third exam.
 - Debrief more on labs. This is what you should have learned.
 - More generally: Highlight key learning outcomes as they happen.
 - Maybe replace the daily writeup with "What were the three main points of this lab?"
 - More time on polymorphism and generics and other OOP stuff. Cut Android to make time.
 - Be clearer on availability.
 - More chance to work from the ground up. But it was also nice to be able to do slightly deeper things because there was an infrastructure. And reading someone else's code is also a valuable experience.

Some things you didn't learn

- Students often put "proficient in Java" on their resumes.
- But you're not. It takes a few years of thinking in Java to be proficient.
- And we also don't cover the full language.
- Nor do we give you deep experience with OOP.
- So maybe you want to take some time and learn on your own

Uncovered (or insufficiently covered) OOP

- More practice with designing objects
- More practice with inheritance
- Key design patterns, including
 - Listeners (is that a pattern)?
 - Flyweights
- And more.

Uncovered Java

- Protection levels
- Swing and other GUI toolkits (I forget what's in Java 8)
- More of the standard classes
- More of the standard utilities
- More event-driven approaches
- Introspection

Uncovered Misc

- Branching in git
- Rearranging commits ("rebasing") in git
- Test suites
- Other build approaches for Java

Preparing for the Final

- Likely to focus more on end-of-semester stuff.
- Do the practice final.
- Look at the take-home final (maybe even sketch answers).
- Make sure that you learn the details of the big data structures.
 - Hash tables.
 - Heaps.
 - Various kinds of lists, including skip lists.
 - BSTs.
- Think about questions I tend to ask (in class, on exams, etc.).
- We can look at a few problems if you'd like.

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.