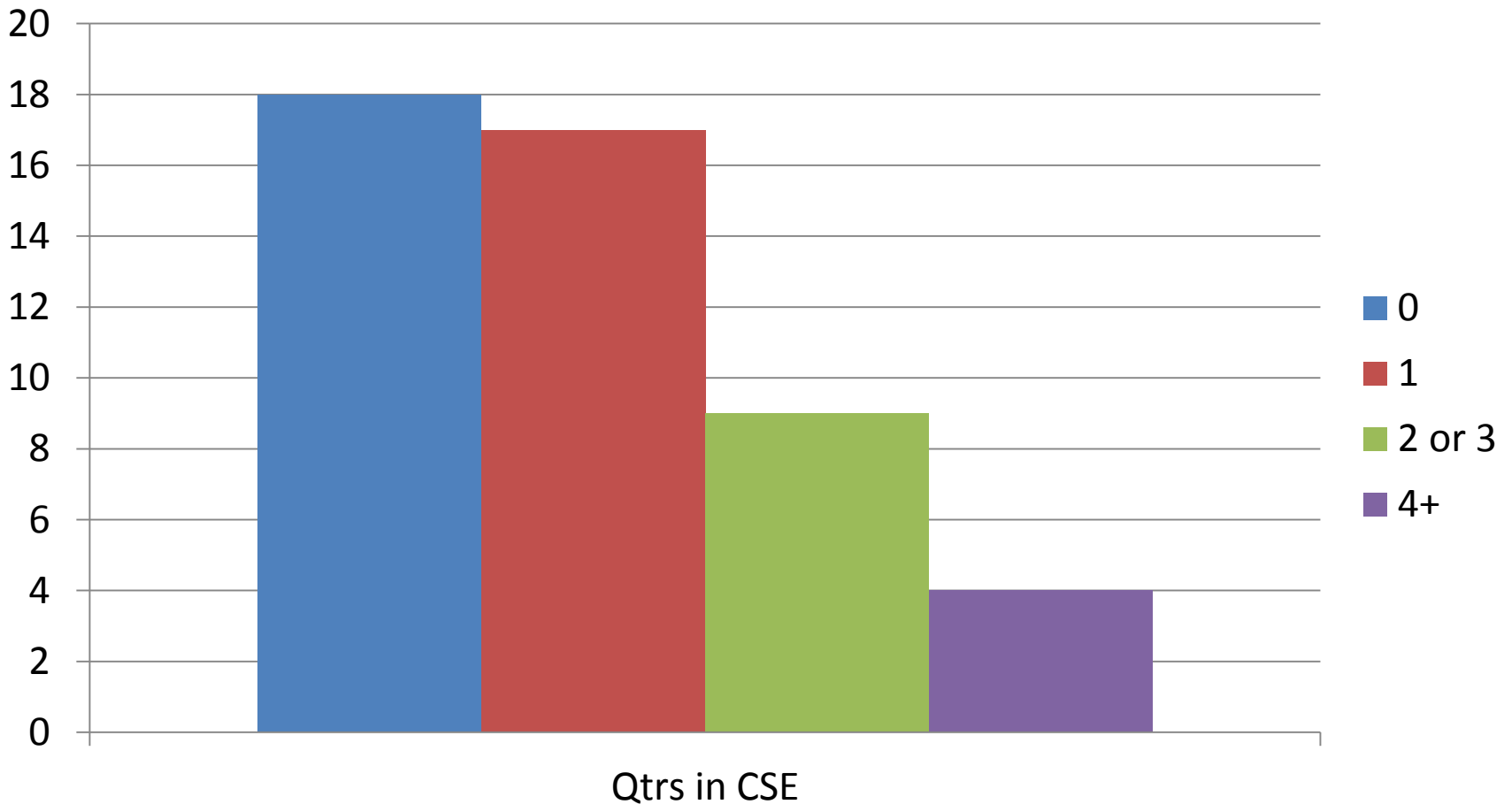
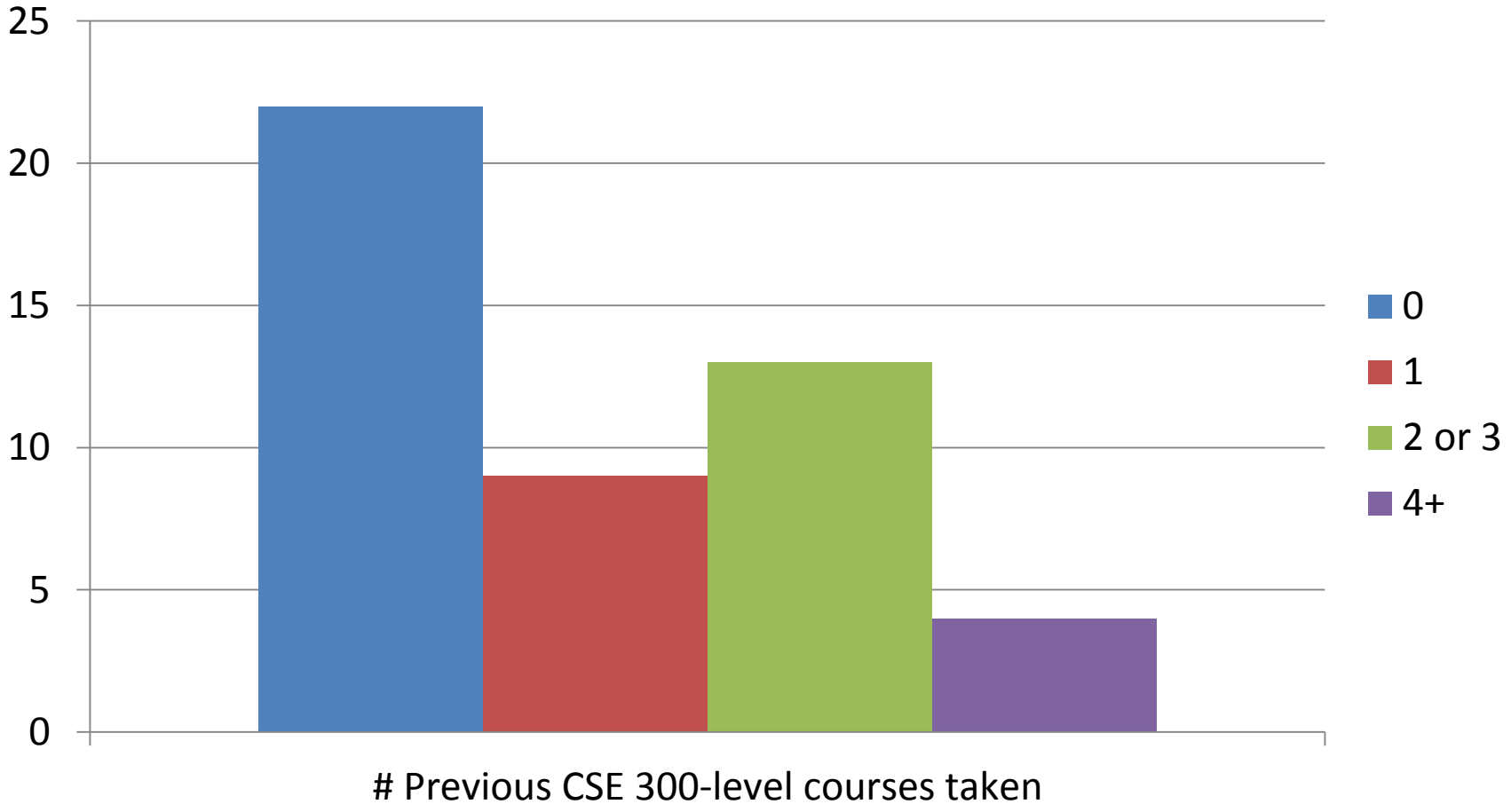
Initial questionnaire: so far 48 submitted

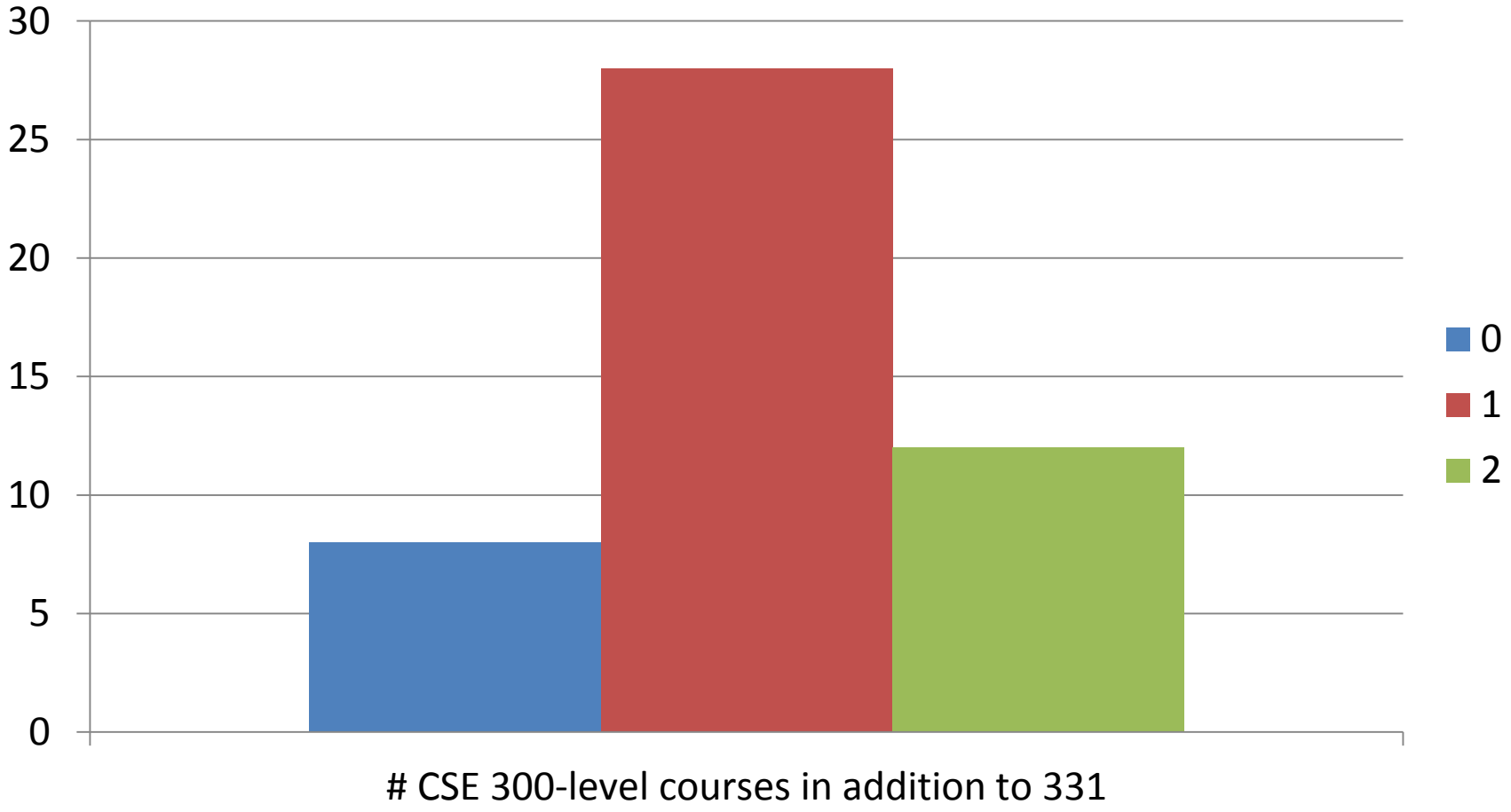
How many quarters in CSE?



Which (how many) CSE 300-level courses have you taken?



CSE 300-levels taking alongside 331?



Gossip about 331

- …a tough but very interesting class.
- There are some group projects, with larger more intricate assignments, done in C++
- took it last qtr, failed it miserably, was uber tough
- a group project, and takes a lot of time each week
- Only what I've read on the catalog. I'm looking forward to the project-oriented assignments.
- It has a lot of programming involved.
- That 332 is more useful :P
- There is a bunch of work associated with this class, but it is totally worth it in the end.
- Tough class with lots of weekly time commitment, although I heard the department was attempting to make the course easier.
- I've heard that it was incredibly difficult spring quarter
- The workload for the class is very high. (20-30 hours outside of lecture).
- Very heavy workload • It is heavy load :) • Heavy workload • Not much except that it has a reputation for being a lot of work. • I've heard it was an unbelievably large amount of work, an MIT semester course crammed into a quarter.
- I have heard that it is a bit difficult.
- That it focuses on software development both individually and in groups.
- I've heard it's a lot of work. I hear people complaining in other classes when 331 deadlines approach.
- My impression is that it's largely about working on software projects in teams, and the language and tool stuff is only a part of that.

Gossip about Notkin

- i have heard nothing, expect so much more than previous professor though
- I've heard he's tough but fair. He also has a gnarly beard, which I've had the honor of seeing in person
- That he has a fantastic beard!

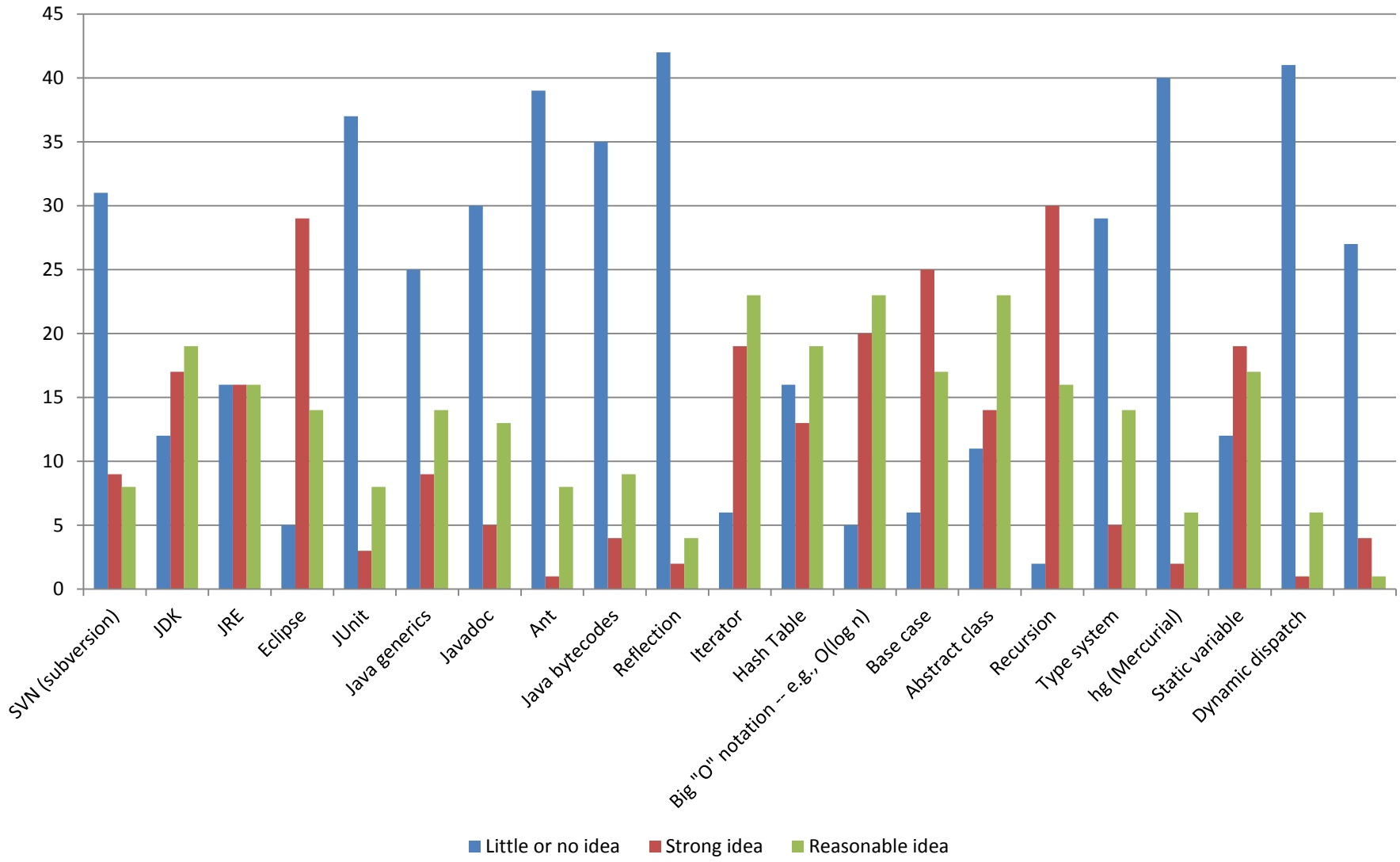
Concerns

- Yes. Since it is my first quarter into the CSE department, I am nervous about my performance. I am even more scared because of my workload …
- Coordinating with another student on a project
- <many workload comments>
- My concern is the language we are going to work on which i guess will be JAVA 6. I am a transfer student and i had all my 100 and 200 level classes with C and C++
- if it is anything like the last quarter it was available, i am in for a world of hurt. dependency between assignments, grading schemes for assignments, grade weights, and breadth/depth of material covered
- workload, and whether I will be able to handle it
- Based on the familiarity questionnaire, it looks like the plan is to do a significant amount of this course's work in Java. I believe we can move faster through learning and applying language-independent concepts with Python because of the elimination of manual compilation and syntactic cruft.
- Not now, but probably once the midterm rolls around.
- Just that the work load is at least semi-comparable to the credit hours
- I'm concerned that I'm rusty, having taken Spring Quarter off for personal reasons. I'm a little worried that it might take time adjusting and that the coursework might move too quickly for me to be able to feel comfortable with the material.
- A concern that I have is that the workload from the previous quarter will not be taken into account when making assignments for this quarter. Hopefully they will be taken into account to allow for a more reasonable workload.
- I'm concerning how difficult this class is.
- Working with other people
- Balancing the workload with CSE 311. Not having taken 311 beforehand. I'm generally worried about my experience since I'm still new to programming and haven't done much outside of 142 and 143 and reading books.

Hope to learn?

- How to improve a given piece of code by modifying its structure and design and how to handle programs involving graphical user interface
- How to successfully collaborate on a coding project and how to effectively use version management depositories etc.
- Project management and the ability to work effectively in a team
- I wish to learn how to be a better programmer and write more sophisticated programs
- whatever will provide me enough knowledge and skills to make it in later courses which depend on this one, and to make it easier for future courses that don't depend on it
- I'm not really sure what the syllabus is, but probably learn more algorithms and approaches to tackle programming problems, and to be a better programmer.
- How to write less code. That is, how to program more cleverly and efficiently.
- What it is exactly that I enjoy so much about software development. Whether it's the problem solving, or the modular (java & classes) design plans... I don't know!
- Master my programming skills.
- I would like to learn how to write effective and thorough unit tests, how to verify program correctness in general including strategies for branch and path coverage, and especially how to work effectively in a team, programming or otherwise.
- How software is developed in the real world
- Something interesting that makes me more knowledgeable about computer science.
- Some of the finer and proper aspects of software design.
- Further develop strong software development skills
- How to work with other people on a single piece of software productively and efficiently as possible.

- I hope to better understand program structure and design, and also to learn more about different programming languages
- How to use knowledge of Java from CSE142/143 to create programs that are useful in real life.
- How to conceptualize large, integrated pieces of software.
- I hope to get a better grasp of the artform behind programming and the knack for working with large and complicated projects, particularly with other people.
- Any thing fun
- Software Engineering techniques and development strategies.
- "A lot … As much as possible…Design, writing good program.."
- All that is intended to be taught in it.
- Hope I could learn something I haven't learned and also group work skills.
- Industry standard unit testing practices and some details of API design and documentation standards.
- Whatever material is presented
- Commonly used techniques in structuring programs
- To be proficient in developing professional looking programs.
- I'd like to learn how to write clear codes and how to make user-friendly applications.
- unsure
- How to efficiently design and implement software whether alone or on a team, alone or on a team.
- I have industry experience so I'm expecting the bulk of the material to be review. But I'm very interested in seeing how this material is taught, and how students new to the material and to software teams approach it.


45
40
35
30
25
20
15
10
5
0
SVN (subversion)
JDK
JRE
Eclipse
JUnit
Java generics
Javadoc
Ant
Java bytecodes
Reflection
Iterator
Hash Table
Big "O" notation -- e.g., O(log n)
Base case
Abstract class
Recursion
Type system
hg (Mercurial)
Static variable
Dynamic dispatch
Little or no idea
Strong idea
Reasonable idea