Transfer Requirements into CS

- Perhaps a little too much stress for those transferring from L&S into CS
- Finding another way to scale the number of students while being more "fair" for all transfers (3.5 GPA in prep classes vs 3.0 GPA in CS 31, 32, 33)

Capstone

- A little unclear as to this aspect, seems like perhaps offering an alternative multi-quarter or more thesis-like project?
- Adding something for CS 130 into the affiliate policy? Making 152B more attractive to students.

Current student sentiments:
  - CS majors are required to take CS 132 if they pick 152B over 130, but 132 isn't really offered that often. (Not sure if this is a cause or consequence?)
  - 152B is considered a large time sink compared to 130 (lack of documentation vs. StackOverflow for 130)
  - Not sure if this is still an issue, but one alumni I talked to explained that the hardware and tools are ancient making build and compile times too long when he took the class (but he also said it was his "absolute favorite" class!)

Lower Division CS Classes (C++ vs. Java)

- In the notes I took for spring, it looks like this also came up in discussion. I definitely still believe that in terms of learning, C++ should be a precursor to Java if given a choice.
- However, one big problem with having C++ as the main language for our lower division classes lies in actually interviewing. Students are put at an inherent disadvantage when using C++ for companies that require coding challenges. I think for interviews that don't actually need code to compile there isn't an issue, but some companies have a coding section during their interviews. Higher chance of timing out from coding challenges using C++.
- Here's a direct quote from an email from a recruiter for one of my on-sites:
  
  Regarding your choice of language, we very strongly recommend you don't use C++. The success rate of candidates using that language is much lower than other dynamic languages. Is there another language you feel comfortable with?

- I also know there is discussion for an additional CS class (in Python?) to give the CS31/32 series more time to explore harder concepts such as lambdas and type inference

Additional Classes

- Further classes in Machine Learning and AI (autonomous in general)
- More elective options or "tracks" like Berkeley's EECS
- Advanced OS class that can be done after CS 111
- More electives such as mobile programming (like Stanford's 193P)
  
  Specifically, one with more in depth topics such as ARC, KVO, Events, etc rather than just what can be learned from a tutorial online.
● More up-to-date technologies (i.e. moving to React/Virtual DOM in CS 144)
● UI/UX classes (will help alleviate people inundating DESMA)

Increasing Alumni Outreach
● This is also sort of a follow-up from last spring's discussion
● Capstone, inviting more alumni and their companies to participate in capstones
● Interviewing, reaching to companies and their alumni to have mock interviews
● Coursera, I remember Ning Ning recommending UCLA to participate more in Open Source ideas like this. I hopefully should have an update by next quarter!