

What should you expect to learn from COS 445?

The first time I taught this course, a student asked me for the five most important things they could hope to learn by the end of the semester. My answer is below, followed by some elaboration in question/answer format - reading it might help you get the most out of the class.

1. **Skill: creative problem solving.** You'll develop this skill by completing the PSets (and midterms/final). Lectures/precepts will help you by going through the process as a group.
2. **Skill: writing clear and convincing arguments.** You'll develop this skill again through completing PSets, processing feedback and comparing your answers to provided solutions.
3. **Skill: reasoning about incentives.** You'll again mostly develop this skill through completing PSets. Precepts will again help by going through the process as a group. Lectures will help by presenting some generic approaches.
4. **Knowledge: specific approaches/results.** Lectures will provide some generic results for reasoning about incentives that are good to have in your pocket. Some examples covered in this class include the deferred-acceptance algorithm, Arrow's Impossibility Theorem, Nash equilibria, proper scoring rules, and the Vickrey-Clarke-Groves mechanism. We'll also try to cover a few core algorithmic concepts like linear programming.
5. **Knowledge: awareness of domains that require reasoning about incentives.** Lecture will expose you to a broad range of "computer science-y" topics where incentives matter.

Which of the five are most important? In my opinion, the three skills are significantly more important than the other two. This means, for instance, that (in my opinion) your time is much better spent engaging with PSets than on additional reading material. For a COS major taking this as their only theory class beyond 340, my personal opinion is that $1 > 2 > 3$ in terms of importance.¹

What makes this class different from 340? You'll be expected to be much more independent. For example, in 340 you probably learned general approaches in lectures, and in PSets practiced applying these tools to specific problems. In 445 it will be flipped: in lectures we'll see specific problems to build intuition, but on PSets you'll need to push well beyond what you learned in lecture to solve harder problems. This is the only way to develop as a problem-solver. The goal of the course is not for you to learn the solutions to interesting problems, but to learn how to solve interesting problems. This is a bit of an oversimplification, but hopefully makes the distinction clear.

Will I need to know lots of math? No, but you'll need to engage deeply with basic probability and calculus. This isn't a math class, and the goal isn't to teach you math. Some problems will require you to be creative with math, but nothing beyond 340 (we'll also provide a cheatsheet and are happy to help with this in office hours).

On the flip side, some problems may be challenging just to phrase as a math question, and you should expect to spend a little bit of effort just to figure out exactly what some problems are asking.

¹But this isn't universal: for instance, if you're a math major taking four classes a semester that grill 1 and 2, probably you should hope to learn most about 3.