What should you expect to learn from COS 445?

The first time I (Matt) 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 primarily develop this skill by completing assignments*. Lectures will demonstrate this skill. Precepts will practice this skill in a group.
2. **Skill: writing clear and convincing arguments**. *You'll develop this skill by completing assignments and processing feedback*. Lecture notes, and solutions provide examples.
3. **Skill: reasoning about incentives**. *You'll develop this skill primarily by completing assignments*. Lectures again demonstrate this skill, and will also provide some general approaches. Precepts again will help you practice this skill in a group.
4. **Knowledge: specific results**. Lectures will teach some key results for reasoning about incentives. Some examples covered in this class include the deferred-acceptance algorithm, Gibbard-Satterthwaite Theorem, Vickrey-Clarke-Groves mechanism, and Nash equilibria. We'll also cover a few core algorithmic concepts like linear programming.
5. **Knowledge: awareness of domains that require reasoning about incentives**. Lectures 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 anything else. For a COS major taking this as their only theory class beyond 240, my personal opinion is that $1 > 2 > 3 \gg 4 \sim 5$ in terms of importance.[1]

What makes this class different from 240? You'll be expected to be much more independent. For example, in 240 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 saw 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 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 some 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 too advanced (we'll also provide a cheatsheet and are happy to help with this in office hours).

On the flip side, many problems *will be challenging just to phrase as a math question*, and you should expect to spend effort parsing definitions just to figure out exactly what's being asked.

[1] But 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.