

What should you expect to learn from COS 330?

This document is a slightly modified version of a document originally written by Matt Weinberg
What do we want you to get out of this class? Why should you take this class?

1. **Skill: creative problem solving.** Tackling problems you haven't seen before: finding the right abstraction, trying small examples, and building intuition before formalizing a solution. *You'll primarily develop this skill by working on PSets.* Lectures will demonstrate this skill. Precepts will practice this skill in a group.
2. **Skill: reading and writing clear, convincing arguments.** Communicating a solution so that someone else (a grader, a colleague, your future self) can verify it's correct – precision and clarity, not just correctness. *You'll develop this skill by working on PSets and processing feedback.* Lecture notes, and solutions provide examples.
3. **Knowledge: tools and concepts to approach problems.** A toolbox of recurring techniques (algorithms, probabilistic arguments, optimization, ...) that show up across many different problems. Lectures will teach you these tools, and PSets will give you practice recognizing when to reach for which one.
4. **Knowledge: great ideas in theoretical computer science.** The landmark results and ideas that shaped the field, and why they matter beyond this course. Lectures will expose you to a broad range of “computer science-y” topics where rigorous reasoning matters.

Which of the four are most important? In my opinion, the two skills are significantly more important than the two knowledge items. 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 $1 > 2 \gg 3 \sim 4$ in terms of importance.

AI can now solve theory problems – what's still relevant? (Disclaimer: this is just my opinion, held at the time of writing. This is a genuinely unprecedented moment, and the field of computer science – and the world more broadly – is still figuring out how to think about it. It's completely fine to disagree with me.)

Being a good computer scientist has always required critical thinking and problem solving, and I don't think that's changed. What *has* changed is that you now also need to know which problems are worth working on, and how to orchestrate AI to help you solve them – and that's itself a skill this course will try to build in you. In this context, *problem solving* just means confronting a situation for which you don't have a ready answer or known procedure – that's exactly the situation you're in when you need to figure out *what* to ask an AI, and how to check whether you should trust what it gives back.

Will this still be true when you graduate? In a year? Honestly, nobody knows. I can picture a world where the answer is no, and one where the answer is yes.

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 330 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 things like basic probability. 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.

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.