A Guide for First-time Course Instructors

Saadia Gabriel
https://saadiagabriel.com/

Introduction. In my first year as university faculty, I got to reflect on how many of the roles we're put into as professors are not roles we're trained for in grad school. One of the most important of these roles is being the solo instructor for a course. During my own time in grad school, I never ran a course on my own, let alone created a new one. Yet suddenly I was in charge of shaping hundreds of young students' futures, and I quickly realized the instructor's side of the classroom is very different from a student's. I decided to put together some of my reflections on this experience into this document, in the hopes it'll be useful to others at the beginning of figuring out what it means to be a good course instructor. I'll start with some pitfalls and successes from teaching grad- and undergrad-level courses, then I'll get into the process of developing a new permanent course on AI ethics.

Teaching as a Black Woman in CS. First, I'll start off with an admission. Teaching was the aspect of being junior faculty I was most scared of when I started the position. I'd been warned by other Black female faculty during my job search that I would be a target. They explained women notoriously get lower teaching evaluations (evidence for this can be found here) and the situation is even worse for Black women.

I braced myself, but throughout my career I've tried to approach my identity as a strength instead of being swallowed up in the feeling that I'm walking into every room at an immediate disadvantage. Throughout my life I've had to deal with race- and gender-based verbal harassment on and offline. But I navigated subtle and sometimes overt discrimination to end up a professor at one of the top universities in the world.

In my first quarter of teaching, I was deeply moved by how appreciative and respectful my students actually were, and how much I learned from their feedback. I think these fears are likely shared by a lot of people who identify outside the stereotype of a CS professor, and I hope that nobody decides not to become a professor because of that. Teaching really can be one of the most fulfilling experiences, and an opportunity to shift perceptions of what it means and looks like to be CS faculty.

Rules of Thumb for First-time Instructors. The next section can be summarized as a "with great power, comes great responsibility" warning, with some experimentally guided rules-of-thumb for navigating first-time teaching. This guide comes with a caveat — it is specific to my experience as a first year professor at a research-oriented large university on a quarter system, so some of this may not generalize to other school systems. For example, our tenure-track faculty generally do not have to deal with teaching multiple classes per quarter/semester. We also don't encourage co-teaching, so all my experiences are as a solo instructor. I'd love to hear about others' experiences and how they may have differed from what I describe.

- ➔ **DO**: borrow shamelessly from others (and credit them)

  There are many excellent lecture slides that have been made public by instructors for intro NLP and other AI courses. It saved me considerable time borrowing slide decks from seasoned teaching professionals like Nanyun (Violet) Peng, Yejin Choi, Yulia Tsvetkov, Noah Smith, Yoav Artzi, Zhou Yu and Mohit Iyyer. I also got help from Alane Suhr and Jason Eisner joining the TeachNLP slack space, with a community of senior and junior NLP instructors sharing knowledge and providing support to each other. I highly recommend joining communities like this, or starting your own with a mix of junior/senior colleagues at your university.

- ➔ **DO**: make sure guidelines are very concrete

  This is especially important for undergraduate students, but all students appreciate clear-cut guidelines. Make sure there is a grading breakdown for each assignment on the first day of class showing how it contributes to their final grade. Make sure you have a rubric for deciding how to grade individual assignments (and deal with regrade requests). If there are any final projects, make sure you have documents explaining the formatting and expectations for success. Also make sure these documents are available as soon as possible, preferably within the first two weeks. I have some examples of final project docs linked on my teaching website (e.g. this one). My students disliked not having a practice exam, but I've also heard there can be criticism if there is a practice exam that deviates significantly from the real one. For undergraduate students, I

encourage making one or having a summary of key details from the lectures/assignments for them to review. Just make sure to note that the exam may include content outside of the review materials, otherwise you will probably get emails!

➔ **DO**: expect conflicts if you have group assignments

Especially if students do not choose their groups, there are likely to be clashes. The most common are around credit, when one or more members of the group drops the ball but still expects to benefit from the efforts of everyone else. Something I've learned from my past experiences dealing with this at UW and NYU is to always require contribution statements on write-ups for group projects. You can then give a base grade to the entire group and extra credit to students who carried an unequal amount of work.

➔ **DO**: make learning active

Active learning has been shown to be critical for improving students' ability to process and retain information, beyond rote memorization. Back in my college days, I took part in an intro CS peer mentorship training program that encouraged us to design and implement active learning modules. Some ideas we came up with included auction games for teaching data structures and a very popular "cupcake lab" for walking through algorithm execution. Some of my favorite ideas from UCLA colleagues include Remy Wang's suggestion to read papers live instead of posting reading assignments ahead of time. This easily generated some of the most lively discussions we've had in my seminars. Many courses also use in-class quizzes to encourage participation. To crowdsource some tricky questions from my intro NLP course I required all the students to leave a question or comment from the lecture in a Google form.

➔ **DO**: schedule guest lectures

I was hesitant at first to add guest lectures to my undergrad course. I wasn't sure if undergrad students would appreciate the more research-focused content in the way grad students do. However, ultimately I think this is a good idea for a few reasons: (1) inevitably there will be points in the quarter when you as an instructor are low on energy (e.g. conference or grant deadlines) and this allows you to recharge while still making

sure the students are learning, and (2) particularly for more applied AI lectures, this can show students exactly how people use and build on the otherwise somewhat abstract concepts they're learning, and (3) it exposes them to what a research career would entail early and presents it as a concrete future option. I've gotten a few follow-up emails from undergrad students who took my course and are now interested in pursuing research projects.

➔ **DON'T**: fail to practice your lectures beforehand

I actually told myself that I was going to practice the lectures. I had a plan in my head to record every single lecture before the quarter even started. None of this happened. As a result, I got very fair criticisms from students that I sounded like I was reading an encyclopedia and I was just repeating exactly what the slides already said (btw, in general you should not just read off your slides during a presentation). No matter how busy life gets, really try not to procrastinate on this. Your future self will be much happier. When I was a college student with severe public speaking anxiety, my thesis advisor told me to record myself, watch the recordings and edit my presentation until I could more or less give it in my sleep. I think this is brilliant advice for lecturing as well. If you are a first-year professor this can be an important confidence booster before going in front of 80-200 students.

➔ **DON'T**: leave borrowed slides exactly as is (personalize)

To avoid the pitfall of reading off slides, it helps to familiarize yourself with the examples beforehand but it can also be helpful to exchange other people's examples for your own. Common concepts and formulas can be left alone, but don't be afraid to personalize the lecture with examples that have meaning for you and that you can tell a story about.

➔ **DO**: be compassionate

Even if you are overwhelmed and drained. Even if it's the 10th student who has gotten a concussion right before the midterm. It's appropriate to ask for medical documentation or for students to go through an official channel like the university accommodations office. But never ignore a student in crisis or belittle them. Reflecting on my first time teaching

senior undergrad students, I think I was unprepared for how much stress they were facing going into the job market and how obsessive they could be about grades. Try to keep their stress levels in mind, reply to the flood of emails about grade concerns in a timely fashion, and make the class feel as predictable a space as possible for a generation facing a lot of uncertainty.

Be compassionate with yourself too. Set boundaries (e.g. “I won’t reply to emails on weekends or after X time of day”). If you fall short of what you wanted to achieve in your first year teaching, this is normal and completely ok. Colleagues joked off my first year teaching troubles as leaving room for improvement in future assessments. I am also appreciative of my colleague Elisa Kreiss’ wisdom that perhaps you should leave space between the end of a class and reading evaluations, to give yourself a break and put yourself into a more objective headspace for processing the information. Meditate, do yoga, drink hot chocolate…and then come back stronger for the next quarter or semester.

Developing a New Grad-level Course. After the success of the grad-level seminar, I decided to develop it into a full-fledged permanent course. There was a lot of demand for my ethics-centered seminars in Winter and Fall 2025, perhaps in part because there were no other AI offerings in the course catalog that mentioned ethics. I am very grateful that I was finally nudged to actually do this by the administrative director of UCLA’s cross-department Data Science initiative (DataX) following a conversation over lunch. I had enormous support from the director (Susan Swarts) and the initiative’s dedicated instructional designer (Clara Choi) in converting my old syllabus into a robust course plan. After getting approval from the CS department, Clara and I spent several months iterating over the syllabus to transform the seminar into a lecture-based format and develop concrete educational goals. Our exceptional department admin helped coordinate the review process from then on. The field chair approved the proposal with minimal revisions and then it was deliberated on in a faculty executive committee meeting. Here’s my rough timeline for starting your own course plan:

1. Start it as a reading seminar, which requires minimal prep of new materials.

2. Gradually create new materials, for example in the second iteration of the seminar I made a few introductory lecture slide decks for each topic in the course.

3. When you're ready to transition it to a full-fledged course, you already have an idea of what worked and didn't work. You can use the reading seminar course catalog description and previous quarter/semester timelines as the starting point for your syllabus proposal. For example, this was mine:

Proposal for Computational Ethics and Large Language Models

Rationale and Objectives:

This graduate course builds on the success of the special topics seminar CS 269: Computational Ethics, Large Language Models and the Future of NLP, piloted in Winter 2025, which attracted 47 students, including Ph.D. students and M.Eng. students. The strong enrollment and positive feedback demonstrated clear demand for a dedicated course examining the ethical and societal implications of large language models (LLMs). The course provides a structured, in-depth exploration of how LLMs are developed, evaluated and deployed, as well as the associated ethical concerns such as data collection practices, privacy violations and societal biases. Students will gain both conceptual and practical understanding of the intersection between natural language processing (NLP) and ethics through lectures, readings, guest talks, and small-scale group projects.

Course Learning Objectives:

- Students will investigate the ethical implications of large language models (LLMs) and how they are impacting various sectors and aspects of society (e.g., healthcare).

- Students will review common evaluation practices and established auditing frameworks (e.g, SMACTR) for ensuring AI safety. They will explore how these approaches apply to novel scenarios and learn to identify potential limitations.

- Students will learn how to critically evaluate data collection practices, with attention to risks, privacy violations, embedded societal biases, and their evaluation processes. Students will get hands-on experience with data challenges (e.g., label subjectivity) through small group data curation and coding assignments.

- Students will learn how to develop and pitch their own AI governance framework in whitepapers presented at the end of the quarter. This will strengthen students' critical thinking and scientific communication skills.

Fit within the Curriculum

This course fills a critical gap in the current graduate Computer Science curriculum by providing opportunities to examine ethical considerations in natural language processing and large-scale AI systems—a topic not comprehensively addressed in existing offerings. While related courses introduce general NLP methods and techniques (e.g., CS 260C, CS 263), none offer a sustained focus on the intersection of ethics, policy, and LLM deployment. The course complements existing courses by equipping students with frameworks for responsible use of technology.

Impact on Other Departments

This course is expected to complement, rather than duplicate, offerings in related departments. It will likely attract students from engineering, data science, information studies, and social science backgrounds who wish to integrate technical expertise with ethical analysis. For example, in Winter 2025, there were enrolled students from UCLA Health and EE. Collaborations with guest speakers and potential cross-listing opportunities can further enhance interdisciplinary engagement.

Course Structure and Pedagogy

The course will be lecture-based, supplemented by guest lectures from domain experts. Students will engage in group projects that culminate in small-scale coding projects using Python, as well as write short perspective papers that explore and evaluate ethical or governance issues in AI.

Course Scope and Variability

While the course will maintain a consistent focus on ethical frameworks and the societal impact of LLMs, specific applications and readings may vary by instructor and/or guest lecturer(s) for the quarter. These variations will remain within the broader learning objectives of ethical reasoning, critical scientific analysis, computational linguistics and applied problem-solving in AI.

Summary

This course transforms a successful pilot into a permanent, scalable offering that meets growing demand for advanced instruction in AI ethics and language modeling. It advances departmental and institutional goals of training technically proficient and ethically grounded AI practitioners who can lead in shaping the responsible future of artificial intelligence.