Teaching Statement

Ryan R. Rosario, Ph.D.

Last Modified March 22, 2024

It has been my privilege and honor to be able to teach at my alma mater UCLA in the Department of Computer Science the past several years. It means a lot being able to be able to give back to the department that gave me so much – practically my entire career. In this document I reflect on my teaching experience since my previous renewal process.

Teaching Philosophy and Style

As an educator, I take a much longer term view of my role and the skills with which my students leave the class. I measure my success not by the grades that students earn but their interest in the material after they leave the class, and reports I receive from students about the course being useful in their internships or jobs sometimes one or two years into the future. I am still in contact with a few students that I taught in 2018, my first term.

In my opinion, students learn best when they can connect the material with something else they find interesting or familiar. When students can show how interesting and impactful the course material is to solving complex problems, students seem to understand the material better. Students also learn best when they can see the relevance to their future careers, and personal interests/projects. Learning is also enhanced when students feel supported by the course staff, and when they feel the course staff is empathetic to their needs both in class and out of class (e.g. diversity and inclusion, encouraging good mental health).

In my teaching, I focus on presenting the important concepts, the building blocks, upon which students build a foundation. From that, they build and create through lecture exercises and discussions, practice problems and graded assignments. I focus on drawing analogies between data management concepts and systems that students already understand, such as human communication and everyday uses of data. Throughout the course, I make use of examples that are relevant to student's lives at UCLA. For example, discussion of UCLA's enrollment system and its various technical challenges is relevant to a course in data management systems, and is something that is accessible to students.

One aspect of my teaching style that I believe is somewhat unique is that I integrate other CS course material into the courses that I teach, particularly data structures (CS 180), operating systems (CS 111), networks (CS 118), and programming languages (CS 131). I do not believe in teaching course content in a vacuum, as that is not how the material is applied in industry; however, I understand the purpose of it as it can be less cognitive load for students. Whenever I draw on material from other courses (e.g. TCP vs UDP, filesystems, lazy evaluation, topological sort), I introduce enough of the concept to students for them to apply it to the relevant course. I also frequently discuss or name drop various technologies that implement the concept being discussed. For example, when discussing streaming systems in CS 143, I name drop Kafka and message queues like RabbitMQ to inspire curious students to research these systems, and for advanced students that may have heard of or worked with these systems. When a student hears a technology they have used, learned or heard about, I feel the student forms a better connection with the course content. This can sometimes give the student more confidence as well. I strongly believe in teaching to the top quartile of the students as I believe that inspires students in the bottom three quartiles to meet expectations with the large amount of support and motivation the course staff and myself provide. I believe that my style is similar to a hike in the wilderness. The leader in the front sets the pace and expectations, adjusting them when necessary. Other staff accompanies, supports and motivates the individuals in the back, so nobody gets left behind. My goal is to keep the highest performing students engaged and interested (rather than bored) while supporting and motivating students at all levels. I believe that this reflects UCLA's goal of excellence and high standards and expectations.

I use lectures as the formal vehicle of instruction. I present an exhaustive set of slides that serve as a supplement (and for many students, a replacement) for expensive textbooks. Using a tablet, I heavily mark up the

slides to expand on certain points, and add additional content that arises from student questions. Embedded in many slide decks are handouts, or exercises, for students to complete during lecture with my guidance. Students receive a special blank copy that is then completed during lecture. Exercises span Bloom's hierarchy from remembering up to evaluating and creating. In the case where a topic is abbreviated, students are provided with additional resources for curious students. I (attempt to) use humor, and also use empathy as a UCLA alum to connect with students to maintain motivation and hard work. I frequently incorporate case studies (stories) from industry about challenges software and data engineers face, how they resolved them, and how the concept we are studying facilitates those processes. When appropriate, I also give a demonstration of how systems work in practice.

Typically, one or two lectures focus heavily on system engineering, critical thinking and engineering intuition. For example, students in CS 143 learn the various technical challenges of working with unbounded, high velocity data. They are asked to synthesize material they may have learned in other CS courses (or use their intuition for those that have not taken a breadth of courses) to develop fault tolerant ways of processing unbounded high velocity data with minimal data integrity errors. This lecture and concept focuses heavily on applying what students have learned in the course and outside, evaluating various methods to solve the problem, and creating their own solution. It is a topic that most students find challenging, but based on industry experience, is a type of thinking that is crucial for students to master as they begin to think about their careers.

Between lectures (on course forums), I frequently post additional content that is less academic in nature. These may be "scholarly" articles (e.g. vetted news/tech media), or blog posts from practitioners whom I feel are trustworthy and have expertise in the relevant concert. I provide this material to

1. Synthesize course concepts between an academic lecture, and how concepts are used and discussed in industry both academically and colloquially
2. Introduce the state-of-the-art and/or timeliness and relevance of what we are discussing

A standard week in my courses consists of the following:

- Two lectures utilizing digital materials (e.g. slides) with follow-along annotations and handouts completed during lecture. There may also be a technology demo.
- Outside support from the instructor and course staff on forums, and in frequent office hours.
- A discussion ("recitation") section where TAs present the material in their own words, and students work on exercises that I create.
- A homework assignment that allows students to practice concepts, apply what they have learned, evaluate various ways of solving a problem and create solutions. Assignments are a mix of programming assignments and "written work." Programming assignments are completed using a Docker container created by the instructor. In some cases, students can resubmit the assignment until they receive a perfect score. This allows students to focus on mastering the material and less on the grade.
- To assist students with different learning styles and provide a record of what was discussed, all lecture materials including the student master copy, the annotations and a recording are provided to students on BruinLearn and YouTube. Sometimes I re-record parts of the lecture if I am not happy with how I covered a particular concept.

Over the years, my teaching methods have changed substantially based on student feedback, and also the COVID-19 remote learning situation. Lectures are more interactive, with a larger diversity of activities, and I try to reach students with different learning styles. I have also reduced the number of slides substantially to facilitate more discussion and Q&A. I also routinely add new course content reflecting the latest developments in the field. For example, in CS 143, I added discussion of streaming systems (2022) and DuckDB (2024) which is a good way to connect students with emerging technologies.

Assessment and Evaluation Philosophy

My assessment philosophy follows from my teaching style and my industry background, and I view it with a highly statistical perspective. Understanding and mastery of material is an unknown latent variable that spans from 0 (no mastery) to infinite. Each course is a window within this range, and each exam is also a window within this range. Assessments should be challenging, and allow students to show deep understanding of the material from simply remembering up to and including creating solutions. Exams, particularly the final exam, are designed so that no student achieves a perfect score, as there is always room for growth and improvement even in the highest performing students. There is also enough space for students that find the course material more challenging to show what they learned, particularly at the remember and understand levels of Bloom's taxonomy. Additionally, from my industry background, I am familiar with the critical thinking skills and rigor that are associated with technical interviews. There is no ceiling to performance and the philosophy is to see how far the candidate can get. I take the same philosophy for exams, and I believe that my teaching and assessment style begin to prepare students for these challenges.

The midterm exam serves as a performance checkpoint and focuses more heavily on the remember, understand and apply levels of Bloom's taxonomy. This allows students to be rewarded for their achievement, while also providing a diagnostic for students that may be below my expectations, or their own. There are usually one or two subproblems that focus on evaluate and create that are designed for advanced students.

My philosophy on the final exam, which is comprehensive and cumulative, is to additionally test high level understanding and integration of the course concepts. Students are asked to combine understanding of multiple seemingly unrelated concepts, apply what they have learned to new situations, and apply course concepts to scenarios outside of data management systems. Test items on the final exam focus much more heavily on apply, evaluate, and create than the midterm exam, while providing students in quartiles 1 and 2 the opportunity to show what they remember, understand and apply.

Some carefully constructed homework assignments are developed to be more vague and ambiguous and require students to think through various scenarios and trade-offs to construct a response. These more ambiguous assignments are graded on completion rather than correctness, with full solutions provided. I would like to move to a model where students can resubmit their work on all assignments until the receive a perfect score. I want to do this to motivate mastery of concepts and decrease stress and concern over the grade. I also believe this may reduce academic integrity issues and place more accountability on the student to complete work honestly.

I maintain a large body of statistics on each exam and assignment, on the items themselves and how outcomes are measured to ensure a high level of statistical validity and reliability.