KEVIN A. ANGSTADT

STATEMENT OF TEACHING PHILOSOPHY

I tend to be suspicious of educators claiming to be successful teachers, which I have seldom observed to be true in practice. Instead, teachers whom peers and students praise tend to be the shining stars of their profession. As such, I make no claims of being a successful teacher in this statement. Instead, I will try to present my past teaching efforts and attempt to characterize my teaching with respect to teaching style, student learning, and inclusive environments.

TEACHING EXPERIENCE

I have been teaching, in some official capacity, for the past eight years. Broadly, I have:

- Taught Computer Science at several levels, including: introductory summer camps; intermediate, major-required courses; and upper-level electives. I also have experience with team-teaching.
- Facilitated learning in small (8-15 students), medium (30-50 students), and large (70-100+ students) classrooms, each for at least two semesters.
- Designed courses from scratch and modified existing courses, including developing lecture material, assignments, projects, and assessments. These efforts include refining the material over multiple offerings of the course as well as incorporating research ideas.
- Managed staffs of graduate and undergraduate teaching assistants (ranging from 1 to 27).
- Participated in professional development programs to improve my teaching (Tomorrow's Professor Today at UVA and the Graduate Teaching Certificate program at U-M) and to help other instructors improve their teaching (serving as an Engineering Teaching Consultant at U-M).

TEACHING PHILOSOPHY

My approach to teaching is exemplified by the following anonymous quote from a student evaluation:

> I decided to attend the first class on a whim and knew by the first meeting that this was going to be a great class. [...] I noticed after a bit that I was applying techniques learned in the functional programming section in my everyday programs. [...] This class was challenging but also a very welcoming place to learn. Kevin was a really great instructor and really made an effort to be accessible to us and engage the class. Thanks again for this phenomenal course!

No document I write will fully capture my teaching philosophy, but I might simplify and summarize my approach to teaching with three interrelated tenets: reaching students who feel left out, active learning and student engagement, and centering course design around goals.

REACHING STUDENTS WHO FEEL LEFT OUT. Such students include—but are not limited to—learners from different fields of study, non-traditional students, and students from different socio-economic or cultural backgrounds. The challenge of reaching these students motivates me to improve and adapt my teaching, and I continue to tweak my content and delivery to reach more students. While designing courses to support students who feel left out, my approach often benefits all.

From my experiences teaching in both the humanities (German as an undergraduate TA) and the sciences, I find that students from one area often struggle in another because they have different

training in the process of learning. Loosely, I have found that some students are more successful memorizing complicated facts and algorithms while others prefer regenerating complex ideas from a small set of ground truths and rules.[1] As such, I design my courses to support both approaches. When teaching students to convert regular expressions to finite automata in programming languages, I both provide the conversion rules (for memorization) and also explain how these rules were created (for rule abstraction). I test students' ability to successfully solve problems that require converting regular expressions to automata, allowing both those students who memorize the rules and those who regenerate the rules to be successful.

ENGAGEMENT AND ACTIVE PARTICIPATION OF STUDENTS. Students learn best when they are actively involved in the learning process. I prefer presenting material using whiteboards (or digital whiteboards) rather than slides as this medium allows me to adapt course material on the fly in response to student understanding and questions. Active learning is a central tenet of my teaching philosophy. In larger classrooms, I often rely on think-pair-share activities, and in smaller settings, I particularly enjoy using techniques such as large group discussions, hands-on technology (coding in the classroom), interactive lectures, and activity-based review sessions. I also include frequent assessment techniques in my courses. For example, I have developed a practice of beginning each of my class meetings with a review of material from the previous classes. I ask students to verbally summarize and paraphrase the concepts they have learned and do not help the students unless they become particularly stuck on a topic. This assessment helps me to gauge student understanding and retention and also incorporates review for exams directly into each class.

GOAL-ORIENTED OVER CONTENT-ORIENTED LEARNING. I strongly believe a teacher's primary role is to be a facilitator of learning. Instead of fixating on a set of topics to cover, I try to design my course material to focus on *problem solving*, *computational thinking*, *process*, and *communication*. Much like novels in a literature class might be chosen to allow for discussion of particular themes and literary devices, I view the topics and facts of a Computer Science course primarily as a vehicle for developing these high-level skills.

My views have largely been shaped by my own undergraduate education in the Liberal Arts, where "lectures" were often driven forward by student participation in response to questions posed by the instructor, and problem solving was a primary focus. As an instructor, I have implemented aspects of this approach to learning in courses that have largely been content-oriented. In addition to the active learning strategies I previously described, I also adapt my syllabi to feature learning objectives and have trained my course staff on goal-oriented exam design (an approach that continues to be used in the class even now that I am no longer an instructor).

SUPPORTING DIVERSITY IN COMPUTER SCIENCE

Our field continues to struggle to attract and retain non-white and non-male students, and I believe that a sense of belonging is a vital aspect of student success. As a member of the LGBT+ community, I have experienced feelings of exclusion and not belonging. While I by no means equate my experiences with those of others, they do help me begin to appreciate the challenges others must overcome to be successful computer scientists. Additionally, I have observed that strong mentorship is common among many successful individuals with underrepresented identities. As a scholar and educator, I consider it my duty to reduce barriers for success and to strive to support all students.

[1] Similar models have been studied in psychology to characterize concept learning (e.g., memorization vs. rule abstraction).

The following are examples of activities I have pursued in an effort to support the broadening of participation in computing disciplines:

- I led the development and facilitation of 90 minute diversity workshops for EECS TAs in collaboration with my department's diversity committee, covering barriers to success (e.g., stereotype threat, impostor syndrome, etc.) and mitigation techniques (e.g., rephrasing exclusionary language, promoting growth mindset, etc.). Over 100 TAs were trained the first semester this workshop was offered. The department is interested in continuing to provide these workshops, and the university's engineering-wide teaching center has also expressed interest in collaborating.
- I acted as the course staff liaison to my department's Computing Cares program, which focuses on supporting students with diverse identities and studying student progress within the department. As part of my involvement, I developed a mini-lecture on stereotype threat and resilience for our data structures and algorithms class that is now part of the the schedule of topics.
- I have applied for and received internal university funding for a mentorship-focused speaker series that invites successful computer scientists from different careers (e.g., liberal arts academia, industrial research, etc.) and backgrounds to speak about their work and advise students. In total, I have secured $30,000 (including matched department funds) to support seven speakers across two years. Over 90% of students taking part in talks and meetings have reported that the series has helped them understand how to achieve their career, academic, and research goals.

Going forward, I am extremely interested in continuing similar activities. Indeed, I wish to pursue a career in which I can prioritize teaching and mentorship of students who might feel as though they do not belong.

TEACHING INTERESTS

I enjoy—and am interested in—teaching computing courses at all levels. My previous teaching, training, and research experience place me in a position to teach courses in:

- Core introductory and intermediate courses, including programming, data structures, and algorithms.
- My areas of expertise: programming languages, computer architecture, and software engineering.
- Theoretical courses, such as theory of computation, discrete math, and formal proof writing.
- Upper-level electives, including compilers, web development and embedded systems.

Additionally, I am interested in developing courses that extend beyond the nominal boundaries of computer science. Topics I am interested in include history of computers, computing and computation through the ages, philosophy of technology, ethics in computing, and music and technology. I believe that it is important for students to both understand where our field came from and also the cultural and societal impact of tools and algorithms they design. Such topics can augment existing interdisciplinary initiatives aimed at stronger cohesion between the humanities and computing disciplines. These topics also provide fruitful opportunities for team-teaching and interdisciplinary collaboration.

TEACHING EVALUATIONS

Copies of my teaching evaluations are available at: https://web.eecs.umich.edu/~angstadt/files/evals/angstadt-evals.pdf.