The topic of the day's lecture was the topological notion of continuity. However, the students already encountered the ε-δ definition of continuity in calculus. I began my lecture by asking the students to recall the ε-δ definition. Then, two students began to debate whether the definition was *for each* ε *there exists a* δ or *for each* δ *there exists an* ε. Deviating from my plan for the day, I told two students to come to the board to explain the definition as they understood it. This led to a friendly, yet lively, discussion over the definition of continuity. I sat down and watched, prompting occasionally to ask them to draw a picture or to consider an example. As a result, the students re-discovered the definition themselves.

The above incident is representative of how I respond to questions in my classroom in a way that allows the students to discover the answers. In the courses that I teach, I encouraged students to think critically about what I am teaching, for example, by asking them to design an example input for an algorithm that I give them or to find a counter-example to a claim that "sounds" right. After enough time has passed, students forget most of the lecture material in a class. However, the retention of materials that a student discovered, rather than was told, is much better.

Teaching Experience

When I teach, I involve the students by asking simple questions, by pausing for a minute (or three) after I present an important theorem to allow the students to digest the statement, and by allowing the lecture to be guided by student-asked questions. I assign homework that balances understanding definitions with synthesizing knowledge. Below, I briefly describe the experiences that I have had while teaching or TAing various classes. Each course has helped me to improve my teaching style.

Calculus. As an undergraduate student, I was a TA and recitation leader for various calculus classes at Saint Joseph's University. In this role, I learned how to redirect a student's goals from homework-solving to concept-learning.

An Overview of Computer Science. I was the lead TA for this introductory computer science class for non-majors taught by Dr. Jeff Forbes at Duke. I was responsible for overseeing the lab assignments, and taught two lectures. I dedicated one of these lectures to teaching the students debugging techniques, as I noticed that many students had difficulty tracing through loops during the labs.

Discrete Mathematics for Computer Science / Discrete Structures. In Spring 2009, I co-instructed this course with Herbert Edelsbrunner. We created a set of Lecture Notes that we made publicly available online. The class was well-received by the students, as was demonstrated in the course evaluations: the quality of the course was rated 4.44/5.0, and encouraging class discussion was rated 4.56/5.0. In addition, the lecture notes that we developed for this course are utilized by discrete mathematics instructors at other universities.

At MSU, I have taught the equivalent course, Discrete Structures, twice (plus a few weeks of a team-taught offering of the class). For an introduction to proof class, this class is large with almost 100 students enrolling each semester. The personalized feedback that I was accustomed to giving students had to be generalized feedback and plenty of examples. A common remark in the course evaluations for this class is that the class "jumps around." Addressing this concern is something I am still working on (and, "trust me, you need simple logic in order to write a proof" does not suffice). One strategy I adopted was to try to motivate the concepts externally. In order to do so, I started to include a "pop math" book as a supplemental book to the class, assigning a few pages of the book per week to go along with the course textbook. My current favorite book to use for this is *Four Colors Suffice* by Robin Wilson.

Topology. Recognizing my enthusiasm when speaking about topology, a graduate student at IST Austria asked me to teach him topology, and I initiated a seminar on point set topology at IST Austria (at the time, no formal classes were offered at IST Austria). We met once or twice a week for 1.5 hours, where I would lecture from Munkres' *Topology.* The students who attended were excited to learn the material, often asking challenging questions. While teaching this course, I became aware of the appropriate amount of material that I cover effectively in one lecture.

Algorithms. I teach both undergraduate and graduate algorithms courses at MSU. The undergraduate course is a senior-level elective course, and the graduate course is a required course for both the MS and the PhD in Computer Science. In both classes, we cover various algorithmic paradigms, including recursion and dynamic programming. We also focus on both the design of and the analysis of algorithms. The two classes differ in the pace that we cover the material and the difficulty-level of the specific algorithms that we cover. In terms of Bloom's taxonomy, in both versions of this class, students must *create* by drawing upon their ability to *apply* previously learned concepts. At the undergraduate level, I support the students by working closely with the TA in providing detailed feedback to all students on their presentation of algorithms and proofs of correctness. At the graduate level, I often have a smaller class (about 15 students) and am able to meet one-on-one with the students to discuss their progress and strengths/weaknesses. Through teaching algorithms, I have become better at supporting in-class group dynamics so that, when I offer a question to the class to discuss in groups, each group is making progress towards answering that question.

Computational Geometry and Topology. In Spring 2013, I co-instructed a small undergraduate Computational Geometry class with Gary Miller at Carnegie Mellon University (CMU). We broke the semester into two main parts: classical computational geometry (convex hulls, Voronoi diagrams, Delaunay triangulations, arrangements, linear programming, etc.) and modern topics (including the computation of the Fréchet distance, approximate nearest neighbor search, and an introduction to persistent homology). One of the first lectures I taught was on oriented projective geometry. A graphics student approached me after the class to let me know that he finally understood some of the computations in graphics after that lecture.

I taught a topics course at MSU on Computational Geometry and Topology in Spring 2016, using the CMU course as a basis for the first half. This course is now two hard-coded classes, Computational Geometry and Computational Topology, offered alternating springs. I have taught Computational Topology three times, and will offer Computational Geometry for the first time in Spring 2023 (previously, another MSU faculty member was teaching this class).

The students who enroll in the Computational Topology class are enrolled in MS or Ph.D. programs in Mathematics, Computer Science, Data Science, and Statistics. As such, I cannot assume a uniform prerequisite across all students: some come in with knowledge of point set topology but no algorithmic background, and others come in having taken courses in algorithms but never encountered mathematics beyond linear algebra. To navigate this, I begin the class with fundamentals of both algorithms and topology, and require students to collaborate across backgrounds on the first assignment. The first time I taught this course, I verbally encouraged them to collaborate, but waited until about six weeks in the class to have them work through a substantial problem in the classroom setting. After this class period, one student told me that she now understood what it meant to work with her classmates. And so, now, in addition to interjecting my lectures with active learning exercises, I dedicate an entire class period to working on a difficult problem early in the semester.

Training

My teaching and mentoring training includes formal training in coursework, participation in university-sponsored workshops, and seeking external training opportunities.

At Duke I received the Graduate Aid in Areas of National Need (GAANN) fellowship while a graduate student at Duke. The aim of this fellowship was two-fold: to train me in both research and in teaching. Funded by the GANN fellowship, I attended the SIGCSE Symposium in 2009 and 2010 (this is one of the two main conferences on computer science education). There, I attended many sessions on teaching techniques and curriculum development. At Duke, I took two teaching classes: Teaching with Technology and Introduction to College Teaching. I also attended various Teaching Ideas workshops, including: Using Acting Techniques in the Teaching/Learning Process, Responding Efficiently and Effectively to Student Writing, and Strategies to Teach Large Enrollment Classes Successfully.

At Montana State University I regularly attend the teaching and mentoring workshops coordinated by the MSU Center for Faculty Excellence (CFE), especially ones focused on active learning. The land-grant mission of MSU provides faculty with additional opportunities to learn about mentoring American Indian students. As such, in academic year 2015–2016, I participated in the *Indigenous Mentoring Program (IMP)*, which a series of nine workshops to help faculty learn how to advise (and recruit) American Indian and Alaska Native students. Then, in August 2018, I participated in the Indigenous Knowledge Field Camp (IKFC) as a substitute mentor. The IKFC is an indigenous cultural immersion experience to examine participant's unique perceptions and beliefs in an effort to recognize the intersections of Native Traditional Knowledge and Western science through a three-day experience on traditional Nez Perce land. IKFC is available to Native American and Alaska Native students and their faculty mentors. Activities included presentations, talks, and individual exploration of stories on traditional practices and ways of knowing.

External Training I have attended several workshops that have helped me improve my teaching and mentoring, including a National Effective Teaching Institutes (NETI) 1B workshop in 2019 and the CIMER Entering Mentoring Facilitator Training just a few weeks ago. However, the one that has had the most impact for me was an improv workshop that I attended in January 2017. facilitated by Katie Goodman. At the time I attended this workshop, I was teaching Graduate Algorithms, and I was struggling with the quality of student's writing. The similarities between good academic writing and good improv acting struck me, especially the 'setting the scene' part: let the audience know what you are doing. Walking the reader/audience through a proof/scene is crucial for the audience to follow the writing/story. In the classroom, I occasionally have the students act out a simple improv game, and use this as a teaching moment to draw parallels to academic writing. This exercise is usually well-received by the students.