The topic of the day’s lecture was the topological notion of continuity ($f : X \to Y$ is continuous if for each open set $V$ of $Y$, the set $f^{-1}(V)$ is open in $X$). However, the students already encountered the $\varepsilon$-$\delta$ definition of continuity in calculus ($f : \mathbb{R} \to \mathbb{R}$ is continuous at $p$ if, for each $\varepsilon > 0$, there exists a $\delta > 0$ such that $|p - x| < \delta$ implies that $|f(p) - f(x)| < \varepsilon$). So, I wanted to assure the students that these two seemingly different definitions coincided. I began my lecture by asking the students to recall the $\varepsilon$-$\delta$ definition. Then, two students began to debate whether the definition was for each $\varepsilon$ there exists a $\delta$ or for each $\delta$ there exists an $\varepsilon$. 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 lead to a friendly heated 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 discovered the definition themselves.

The above incident is representative of the topology seminar that I lead at the Institute for Science and Technology Austria (IST Austria). In this seminar (as well as in other courses that I have taught), I encouraged students to think critically about what I was teaching, whether it is a hard theorem or a definition. 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. I strive for an active learning classroom. In theoretical computer science, an active learning classroom is one that encourages the students to attempt a proof before listening to someone else present it, or to ask questions about transposing the order of for each and there exists in a definition.

Teaching Experience

When I teach, I involve the students by asking simple questions, by pausing for a minute 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 complement the lectures with 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.

**Computational Geometry.** In Spring 2013, I co-instructed a small undergraduate Computational Geometry class with Gary Miller. 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, the 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.

**Topology.** Recognizing my enthusiasm when speaking about topology, a student at IST Austria asked me to teach him topology, and I initiated a seminar on point set topology 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.
Discrete Mathematics for Computer Science. 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, several of the students commented that I was easily accessible outside of class. I have recently become aware that the lecture notes that we developed for this course are utilized by discrete mathematics instructors at other universities.

An Overview of Computer Science. In preparation for teaching a future course (Discrete Mathematics), I observed teaching techniques in this introductory computer science class. I also acted as the lead TA for this course and was responsible for overseeing the labs that accompany this class. In addition, I 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 and using a debugger.

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.

Mentoring

Along with Carola Wenk, I co-advised an undergraduate senior capstone project last year. Currently, we are co-advising a graduate student in Mathematics.

Additionally, I have worked closely with graduate students during my three years as a postdoc. Mahmuda Ahmed and I have implemented the LH distance described in my research statement. Fabrizio Lecci and I have developed an R package which implements techniques in statistical topological data analysis. As co-instructor for Computational Geometry at CMU, I mentored three of the five course project teams.

Training

I received the Graduate Aid in Areas of National Need (GANN) 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: Evaluating Critical Thinking, Using Acting Techniques in the Teaching/Learning Process, Responding Efficiently and Effectively to Student Writing, and Strategies to Teach Large Enrollment Classes Successfully.

Teaching Interests

My experience has prepared me to teach a variety of computer science courses, including, at the undergraduate level, discrete mathematics, data structures, and algorithms. At the graduate level or as a topics course for undergraduates, the courses I am prepared to teach include computational topology/geometry, algorithms, and data analysis classes. Additionally, I would be interested in teaching a non-majors course focusing on using computer science to solve problems in other disciplines.