# Teaching Statement

*Bryan Mills*

*Department of Computer Science, University of Pittsburgh*

*bmills@cs.pitt.edu*

I've had the good fortune of having teachers and mentors encourage me to seek a deeper understanding of the world and equally important motivated me to share my understanding. As a result I believe teaching is an essential part of being a scientist, there is little gained by knowing the answer if you are unable to explain it. This desire to both understand and explain the world has led me to pursue a career in academics.

I have had a range of teaching experiences including being an adjunct professor, teaching assistant, advisor to undergraduate students and manager of recent graduates. This range of experience culminates into the following teaching concepts:

- Make yourself available to your students.
- Challenging your students requires you to be prepared.
- Provide opportunities for students to have hands on experiences.
- Be passionate about the subject matter and the students.

As an educator I always make myself available and encourage my students to ask questions and challenge them to find what is interesting to them. Being available means your never too busy to answer a question in the hallway or have a discussion about the upcoming exam. Just as important, I want to teach in an organization that values this and encourages education beyond the classroom.

The process of challenging students causes me to stay curious and agile, forcing me to be prepared for questions which I might not know the answer. In my teaching experiences I have several examples of students asking questions that challenged me and pushed me to learn more about the subject material. Its a great feeling to know that I've taught students well enough so that they can ask difficult questions, and I also know how rewarding it is for the student when they ask a difficult question.

I strongly believe that "students hear, but may forget," but when "students do, they usually understand." In teaching, I have a strong commitment to developing challenging projects that provide the students with an opportunity to experiment with the basic concepts discussed during the lectures. I have found the process to be rewarding for both the students, who usually savor the satisfaction of accomplishment when the assignment is finished, and for the teacher watching them bring a unique set of talents to solving these problems. Its also important to schedule project deadlines such that you are readily available to answer questions and support the students in their learning. The process of working on fairly large projects within small teams is especially important as a means to prepare students for employment or graduate school.

Lastly, being passionate about the subject matter is critical to inspiring students to learn and go beyond the classroom with their knowledge. I'm routinely accused of getting too excited in the

classroom, but I think the excitement rubs off on those around me and forces the students to pay attention, even if they didn't want to. Equally important is showing that enthusiasm for students and their ideas, which means encouraging students to explore their thoughts and pointing them towards material of interest. For example, I've had students show interest in a topic that was not originally covered during the course and I've had student work on projects in that area and present their findings to the class. This worked as a great way to motivate students and encourage them to contribute directly to the course material.

## Experience

- Taught courses in networks, distributed systems and artificial intelligence as an Adjunct Professor at Bethany College.
- Taught an introduction to computer programming course at the University of Pittsburgh.

## Classes I would like to Teach

**Introduction to Distributed Systems**. Distributed systems are now a cornerstone of most computing environments therefore I believe it is essential that these concepts be taught at a undergraduate level. I would envision this class being split into two equal parts the first being a survey of legacy systems and the second looking forward to next-generation systems that use tools like job queues, map-reduce and other cloud-based services. This is because recent graduates in computer science will be working in a distributed computing environment, from web development to mobile development and they will need these concepts to succeed. I envision being offered as an undergradate course.

**Parallel Algorithms and Programming**. Learning how to design and construct parallel algorithms and implement them is essential in the era of Big Data. Students will learn how to analyze algorithms in terms of computing, communication and synchronization time. This class will review and explore concepts of processes, threads, locks and multi-process safety. Additionally, this course will provide a working understanding of different parallel programming languages such as NESL, MPI, OpenMP and more generally functional programming languages. This course would be targeted at graduate students although it would be approachable to advanced undergraduate students.

**Scalable Data Analytics**. Throughout my professional career, I've been asked to scale analytical processes to either large datasets or large number of users. I would like to develop a course that would teach some of the key concepts I've learned as well as draw on recent research in big data analytics and data driven science. Average computer users have grown to expect immediate responses to difficult queries and this course will teach enabling concepts like job queues, caching, query optimization, alternative database schemas, OLAP systems, object databases, partial responses and inexact solutions. This course would differ from the other two courses because it would focus upon implementation details and scaling concerns of existing algorithms. This course could be taught as either a graduate or undergraduate course, with additional focus on research at the graduate level.