

Teaching Statement

Jinzhong Niu

December 19, 2009

The opportunities to teach and work with students and to develop new educational materials and techniques are among my primary reasons for seeking an academic career. My academic background, teaching experience, and my experience in software development and project management make me well-prepared to teach introductory computer science courses as well as courses on operating systems, distributed systems, artificial intelligence, data structures, and algorithms. My research on agent-based computational economics enables me to teach more advanced topics, including multi-agent systems, reinforcement learning, evolutionary game theory, e-commerce, and distributed systems.

Teaching Experience

My teaching experience started when I was a graduate student in computer science at Beijing University of Aeronautics and Astronautics. I assisted in teaching the undergraduate core course *Operating Systems* from 1996 through 1998. To provide hands-on experience with operating system concepts to CS majors, I initiated a mini-course called *Operating Systems Experiments*. I designed projects based on MINIX, an instructional operating system written by Andrew Tanenbaum, to expose students to process scheduling, memory management, file management, and other fundamental mechanisms in a practical system.

In 2003 and 2004, I taught undergraduate core courses, including *CSc 33200: Operating Systems*¹ and *CSc 31800: Internet Programming*,² at the City College of New York as an adjunct faculty. This was a great experience because I was able to explore every aspect of running a course, including creating a syllabus, choosing a textbook, writing and presenting lectures, developing assignments and supplementary materials, designing and guiding laboratory exercises, and grading assignments and exams. As another example of my initiative, the syllabi I created contained heavy components of course projects. In CSc 33200, I designed projects based on NACHOS, a different instructional OS born at UC Berkeley, for students to experiment on SUN workstations, and in CSc 31800, I asked students to build a simple web server step by step through the semester.

In January 2005, my work experience with Linux and teaching experience with operating system courses won me a contract to teach a week-long course at the Working Connections IT Faculty Development Institute of the City University of New York. This course aimed to expose CUNY community college faculty to the usage and administration of Linux and help them pass on their expertise to their students. Some of these faculty students keep in touch with me and ask me for suggestions on their courses even to the present.

Most recently in Spring 2009, I assisted Professor Simon Parsons, my Ph.D. advisor, in teaching the doctoral course *CSc 84200: E-Commerce and Computational Economics* at the Graduate Center, CUNY. I lectured on the Trading Agent Competition on Market Design, or the CAT Game,³ which has been organized

¹<http://www.cs.gc.cuny.edu/~jniu/teaching/csc33200/csc33200.html>.

²<http://www.cs.gc.cuny.edu/~jniu/teaching/csc31800/csc31800.html>.

³<http://www.marketbasedcontrol.com/cat/>.

annually since 2007 to promote research on agent-based e-commerce. I also guided the students to team up and design market making strategies for CAT as their course project, and organized a mini CAT competition⁴ to evaluate their strategies. One team from the class participated in the 2009 CAT Competition and finished in second place in the final.

Teaching Philosophy

My teaching experience led me to view the role of a teacher as the bridge between two worlds, the body of scientific knowledge and the eager student learner. These two worlds are constantly changing and the teacher is called to establish a healthy relationship between them that will eventually continue on its own. To accomplish this goal, the teacher must *clearly present* relevant materials and *engage* students to the subject, must *excite* students about the coursework, and must make sure that the student learns not only *what* is out there, but more importantly *how* to reach towards it.

My teaching experiences have shown me that each student learns differently, so clearly presenting material requires effort beyond simply knowing the material. Some students learn best through hands-on projects, others through attending lecture, completing homework, reading the book, or asking one-on-one questions during office hours. Good courses require interesting and relevant projects and homeworks, high-quality books, understandable lectures, and answering one-on-one questions with patience.

Example isn't another way to teach, it is the only way to teach.

——— *Albert Einstein*

A big challenge in teaching is that students get bored easily with theory and like to see more practical examples. I respond to this challenge with multiple strategies. First, I keep the presentation of theory as short as possible, and spend more time on examples. Second, I get benefit from graphs and demonstrations. Third, I get my students involved in discussions by asking them questions on the subject, or by having them face challenging concepts and sometimes paradoxes. Finally, since, for instance, my classes at CCNY were diverse with students from different regions, especially the Middle East and Asia, where people usually emphasize theory, I encourage the class to form multiethnic groups on an ad-hoc basis in order to promote teamwork which helps students view problems from different perspectives.

When work is a pleasure, life is joy. When work is duty, life is slavery.

——— *Maxim Gorky*

Technically speaking, teachers teach and students learn, but teachers are not the source, nor are students the drain. Teaching and learning will not be effective unless teachers are enthusiastic and students are proactive. It is part of teachers' responsibilities to lead students to think that the subject is interesting and fun. A student who is curious and interested in the subject is easy to teach, but unfortunately not all students arrive in the classroom in this state of mind. To arouse students' interest is a challenging task and needs extra effort. One example I would like to give is an assignment that I devised to teach the concepts of abstraction and encapsulation in object-oriented programming languages. I wrote a class, made only its interface known to the students, and asked them to write code to access the class and retrieve information. To make the task interesting, I encoded in the class a mapping from student names to numbers, and asked students to retrieve their exclusive secret numbers. The task for every student was unique, which obviously excited my students.

⁴<http://146.245.250.181/cat/viewer/?game=1#actu>.

I knew that some of them remembered it the next day when they greeted each other with ‘did you get your number?’ rather than the usual ‘what’s up, dude?’.

I hear and I forget. I see and I remember. I do and I understand.

——— Confucius

This example also demonstrated my emphasis on hands-on experience in teaching. One of the most effective ways I have found to teach students computer science concepts is to let them ‘touch’ and ‘feel’ those concepts through laboratory exercises. When I was an undergraduate student in computer science at Shandong University, I took several semester-long project design courses, each following a theory course. I benefited a lot from this curriculum. I incorporated similar arrangements into those courses I taught at BUAA and CCNY, and the results were very gratifying. My students warmly welcomed this approach since concepts and algorithms in textbooks were no longer dull words, but recipes that they could follow in their practices and souls that they perceived in the real systems.

Give a man a fish and you feed him for a day. Teach him how to fish and you feed him for a lifetime.

——— Chinese proverb

Finally, I believe the most important lesson one can learn is *how* to learn. Facts, techniques, and algorithms often grow outdated. If students fail to learn the importance of lifelong learning, they become lost as their knowledge and skills grow outdated. Computer science is a young field which evolves quickly, so keeping up with the field requires constant learning. As no clear way exists to teach someone how to learn, my approach is to challenge and encourage students to think critically. I emphasized, for example in the CSc 33200 course, *how* and *why* the Internet became what it is, rather than *what* it is. With a habit of critical thinking and by remaining up to date myself, I can prepare the students to answer not only the questions we pose to them today, but questions we have not yet imagined.

Face-to-face instruction has continually challenged me to make lessons fresh and effective. Teaching computer science to students is exciting in its demands and personally rewarding. It has improved my research and writing, as I review and present topics in different ways, with broad perspectives, and to new audiences. I teach, and I learn. Isn’t this yet another instance of ‘I do and I understand’?

As I am constantly trying to become a better teacher, I also realize more deeply the great responsibility that comes with it. It has been said that a teacher plants seeds in the students’ souls. The teacher may not live long enough to see the fruits of that effort, but the students’ lives are inevitably touched forever.