

Teaching Statement for Tudor A. Dumitraş

I believe that the most important job a teacher can do is to help students discover the beauty of computer science and to inspire in them a passion for knowledge. My own interest in fault-tolerant distributed systems grew out of a project-oriented course that I took at the École Polytechnique in Paris. Implementing a group-communication system based on failure detectors (an exclusively theoretical technique, at the time) seeded the knowledge and curiosity that form the engine of my research activities.

From my experience, and from the training I sought at CMU's Eberly Center for Teaching Excellence, I learned to combine active learning and hands-on assignments in my teaching activities. These skills are equally important when advising students on research. As an undergraduate student, I guided lab work for a class in electromagnetism. At CMU, I was a teaching assistant (TA) for two graduate-level courses (on distributed systems) in the ECE department and one undergraduate-level course (introduction to computer systems) in the CS department. For this last course, my students gave me an evaluation of 3.93 out of 4.

My most effective way of conveying knowledge directly is to engage students in active discussion. This works for both graduate and undergraduate students. For example, as the TA for a Ph.D.-level course on dependable embedded systems, where students debate controversial topics (e.g. N-version programming), I helped the instructor lead the in-class discussions. Illustrating such topics with industry examples helps clarify the underlying concepts, and I learned to draw from my own practical experience, acquired through four internships. Similarly, as a TA for the 150-student, sophomore-level "Computer Systems: A Programmer's Perspective" (CS:APP) course, I encouraged lively discussions during my recitations. This interactivity mitigated learning problems and even attracted students from other sections to my recitation. I also recommended a few accessible research papers to the highly motivated students; for example, I used a survey of the Therac 25 accidents to emphasize the impact of race conditions. Five of my CS:APP students later joined my research group, and one of them became a Ph.D. student.

However, in the CMU tradition, I believe that students learn mainly from what they do. Good system design, in particular, can only be taught through hands-on experience. I guided my students through the implementation details of class projects in both CS:APP and a graduate-level course on fault-tolerant distributed systems. In this course, I also tried to convey my passion for empirical science by giving a guest lecture on the experimental evaluation of distributed systems and by involving the student experimenters in a conference publication. Such hands-on experience shapes critical thinking and fosters problem-solving skills.

Most of all, I enjoy advising students on research projects. There is no precise recipe for teaching creative thinking. The only tool we have for nurturing great scientists and engineers is our guiding influence over what our students do to learn. I was fortunate to work with two great undergraduate students and to contribute to their first publication. I also advised two MS students on their research projects. I truly enjoy the process of helping people learn, as well as the amazing education that one gains while teaching others.

Given my research background, I am qualified to teach introductory courses in programming and computer systems, as well as graduate-level courses in distributed systems and fault-tolerance. In particular, I am interested in teaching courses with a strong hands-on component, where students learn by building interesting systems and by evaluating them. Moreover, my experience showed me that, while grades motivate the average students, competition motivates the truly exceptional ones. These students learn quickly and seek challenges. I would like to explore opportunities for inspiring their excitement about Computer Science in other students. For example, I would like to introduce a competition between two universities (e.g., between undergraduate students from courses that rely on the same textbook).