Steven A. Wolfman
Teaching Statement

December 8, 2003

Every great teacher I have ever known believes with unshakable firmness that his or her subject is the most exciting, engaging, and fascinating in the world. Most have true respect for the vast diversity of fields and ideas, but get them on their own subjects and their eyes widen, their pulses quicken, and their audience cannot help but ride the tide of enthusiasm for the moment and, occasionally, for life. Students that catch this tide transform into self-motivated learners, creating their own knowledge through avid exploration. When I can raise this tide for every student — make every student a self-motivated learner — I will have mastered teaching.

My enthusiasm for computer science and for learning has always been a given, and I try to communicate that enthusiasm in my classes. In part, this means exposing my excitement through my voice and gestures. My students' anonymous evaluations show that they recognize this: 91% of my students across the two courses I instructed gave my enthusiasm the highest rating: "excellent."

Communicating my enthusiasm also means understanding what is exciting about each topic we discuss. One topic proved particularly challenging: #include and function prototypes in introductory C programming. Among local instructors, it was a notoriously boring topic. I felt that if I could not at least express why this topic was important — and preferably why it was interesting — I should not be teaching it. So, I worked with the material until I had developed two complementary stories to motivate it: (1) solving a daunting problem, that of resolving declarations for mutually recursive functions, and (2) introducing the software engineering principles of information hiding and encapsulation. These perspectives prepared me to be enthusiastic about #include and function prototypes and also tied them into the broader picture of computer science for my students.

With experience as a teacher, I have learned that my intuitive enthusiasm for computer science is necessary but not sufficient to raise the tide of enthusiasm for my students. Since each student builds understanding of computer science on her own individual foundation, cultivating different students' enthusiasm demands presenting many different perspectives on computer science. Because of the diversity of students and the complexity of learning, managing these perspectives requires regular communication with students.

One early mistake made this lesson clear to me. In the first class I taught, I assumed incorrectly that my students had been exposed to Unix and allocated no time to teaching it. Fortunately, a homework writeup with opportunities for student feedback revealed that most of my students had never touched a Unix machine. If not for that coincidentally well-timed writeup, my students' enthusiasm would have evaporated in the struggle to cope with a challenging new operating system. This serendipitous feedback has taught me to maintain many channels of communication.

Since then, I ensure that my classes include frequent communication opportunities. All homework assignments include written feedback like the one that saved me from a Unix disaster. I

frequently ask questions of students in class and perform active learning exercises. For example, I started most class sessions in my introductory programming course with a discussion question, like the "coins and scales" problem[1] for a session on conditionals. I ask the university's instructional development group to solicit student feedback midway through the term. I also build community in the class with a few social events such as a data structures-themed Star Wars viewing.

A shift in my research focus toward educational technology has brought a third dimension to my teaching philosophy — besides intuitive enthusiasm and lessons of experience — which is founded in the vast body of research on pedagogy. Resources such as the National Research Council's report "How People Learn" [1] or Felder's work on learning styles [2] validate the teaching techniques I use and inform my efforts to articulate and improve my teaching strategies.

For example, understanding pedagogical literature has advanced my efforts to teach to a diverse group of students by describing ideas from multiple perspectives. I began teaching this way on the advice of my high school physics teacher. The constructivist theory of learning — in which people build new knowledge by linking it with existing knowledge [1] — legitimized and clarified this approach for me. From a constructivist standpoint, teaching from multiple perspectives promises better learning by connecting to more students' background knowledge. Later, research on learning styles [2] helped me understand how different *styles* of teaching (not just different perspectives) are important. The learning styles work, in turn, inspired me to contribute back to the body of pedagogical literature by creating a repository of physical or "kinesthetic" classroom learning activities, which I and two colleagues will present at the SIGCSE computer science education conference.

I strive to fold all these influences — intuition, experience, and research — into my teaching strategy. With time, that strategy has grown from a desire to motivate students into a richer appreciation of how and why people learn. I eagerly anticipate expanding my teaching strategy further with help from students, colleagues, and scholarly resources. Above all, I always remember how *unbelievably* exciting computer science is, and I watch with joy as *my* students catch that irresistible tide of enthusiasm for the field and for learning.

Teaching Interests: I have the experience and enthusiasm to teach a broad spectrum of computing courses. My teaching experience prepares me particularly well for introductory and core computer science courses. My research experience qualifies me to teach advanced courses in human-computer interaction, educational technology, and artificial intelligence. I am particularly excited about infusing user-centered and iterative design into such courses. With time to refresh my electrical engineering background, I can teach undergraduate hardware courses such as circuit design. Finally, my outreach and diversity efforts with my research group prepare me to teach courses targeted at attracting diverse groups to computer science and interdisciplinary courses with related fields such as education or psychology.

References

[1] J. D. Bransford, A. L. Brown, and R. R. Cocking, editors. *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C., 2000. Expanded Edition.

[2] R. Felder and L. Silverman. Learning and teaching styles in engineering education. *Engr. Ed.*, 78(7):674–81, 1988.

[1] You are given three coins, but one is fake. The fake one is heavier than the others. Using a balance scale, how many weighings do you need to find the fake?