Teaching Statement
Richard C. Davis

A teacher plays a vital role in helping his students discover their gifts and the confidence to apply them to problems they find compelling. By observing how my instructors taught me, I have seen that teaching self understanding and self-confidence requires more than showing students how to solve challenging problems. Many students need to be shown how computer science can be applied to problems that interest them before they will be motivated to learn more. Also, by working on team projects in later classes, students will begin to see how their own interests and creativity enables them to collaborate with others. Finally, by involving students in ongoing research projects, they will develop a belief that this experience can extend beyond the classroom. Here, I will begin by describing a particularly positive learning experience in an undergraduate digital design class. I will then show how experiences like this one have helped me teach self understanding and self-confidence to engineering students.

My experience of computer science and engineering was rather humdrum in the beginning of my undergraduate career, but the instructor of my digital design class gave me my first opportunity to build anything I wanted. I pursued my own curiosity about music and instantly discovered that I was more engaged and excited than ever before. I chose a teammate who was a trained musician, and with his knowledge of keyboards and digital music and my vision, we made a formidable team.

My teammate and I planned a system that played music with sampled audio, but our design was overly ambitious. Fortunately, our teaching assistant was patient and helped us pare it down to two components that fit our skills and interests perfectly: a keyboard interface component for my teammate and an audio sampling component with pitch detection for me. In order to learn about pitch detection, my TA also suggested that I talk to a researcher that he knew. Through my interaction with that researcher, I learned many approaches to this problem, and saw an opportunity to continue working on his research project. At the end of the semester, I had built something I was proud of, but more importantly, I left with a belief in my own ability to tackle interesting problems. My teachers helped me to connect the subject to my own interests, showed me how to extend my reach through collaboration, and helped me gain confidence that I could continue work in this area through research.

The first step illustrated in the above example is getting students to connect computer science to their own life experience. Students must see that the material they are learning empowers them to tackle problems that interest them. When students believe this, they motivate themselves, and it is much easier to prepare them for their future. Helping students to see how computer science connects to their lives is especially important in introductory classes, where some students are still searching for these connections. For example, I taught a required class at the University of Washington that introduced Computer Science undergraduates to Unix shells, scripting, and C/C++ programming. Many students in this class struggled to understand why the content was important. I surveyed students in the first week to discover their areas of interest and designed homework assignments to show how the course material would help them work in these areas. In the final course survey, students were much more positive and said that these homework assignments were very helpful in preparing them for their future.

Extra-curricular classes are another excellent way to connect students with computer science. Through MIT's Educational Studies Program, I had the opportunity to teach two ten-week extra-

curricular courses to teenagers. The first was structured around making video games, and the response was phenomenal. I used the opportunity to introduce a number of concepts from data structures, to graphics, to real-time programming. I could see that many of my students had stopped thinking that creating games was just a dream and began motivating themselves to figure out how they could make it happen. The second course focused on 3D art, which caused me to add more linear algebra and reduce the focus on programming. This attracted a different demographic of students (*e.g.*, more women) but also produced many self-motivated learners.

My undergraduate digital design experience also illustrates how team projects can build self understanding and self-confidence. By working in teams, students gain confidence and develop skills that are necessary for building large computer systems. They can also come to a deeper understanding of what makes them unique and learn how their talents contribute to a team. When observing students at work on projects, a teacher may encourage a student to explore their special talent for expressing a vision, managing a team, or writing code. No team will have a perfect balance of talents and styles, but shepherding students into more balanced teams may help their strengths to emerge. In particular, it is good to involve non-CS majors whenever possible, as they can often bring in a great deal of experience with a particular problem domain or artistic skills that can balance out the technical skills of the CS majors.

As a teaching assistant, I was a guide to undergraduate project groups in both a graphics course and a human-computer interaction course. The HCI course was particularly exciting, because students collaborated throughout the quarter, and because most teams included one member from the school of design. I arranged students into teams and shepherded them throughout the quarter, and I was delighted to see many students develop strong development, design, organization, or presentation skills. It was clear that most students were personally invested in their projects. Many expressed new found confidence in their ability to contribute to a meaningful software project. Industry representatives who attended the students' final presentations were amazed at the quality of several designs. One team designed a cell phone-based interface for real-time bus information that was successful enough to continue as a research project after the course concluded.

The last concept illustrated by my undergraduate digital design experience is the importance of research to students. Involving undergraduate students in ongoing research projects is an excellent way to cement their confidence and skill. Managing workers who are in the process of acquiring skills is a bit of a challenge, but lack of skill does not always make it necessary to turn away an interested student. In my NotePals and K-Sketch projects, I have managed several undergraduates who did not have the skills necessary to work directly with the program code. Instead, these students made meaningful contributions to user studies, web sites, and documentation. In situations like these, it is important only that students have an opportunity to exercise their own creativity, to see how it affects the project, and to observe the impact that the project has as a whole.

Graduate students, on the other hand, need research in order to define their career themes. A good mentor is vital for students to discover their strengths and learn how to apply them to the problems they are most passionate about. When the pressures of graduate school become overwhelming, the mentor's perspective is necessary to help students prioritize tasks. Though I have never served as an official advisor or mentor, I am proud to say that I have played a significant role in advising three students who were struggling to define their themes and were close to giving up on their graduate degrees. All of these students are now well on their way to promising careers.

I believe that the teaching methods I have outlined here will enable me to help many students identify their strengths and find the confidence to contribute to important real-world problems. By showing students how computer science can be applied to problems that excite them, I will help them become self-motivated learners. By encouraging them as they work in teams, I will help them find their gifts. By giving them opportunities to contribute to research projects, I will help them find the confidence to take their gifts to the outside world.

My research in Human-Computer Interaction and Computer Supported Cooperative Work has equipped me well to teach a broad range of undergraduate and graduate classes in this area. Because my work requires the construction of complex software systems, I also have the ability to teach undergraduates a broad range of applied computer science courses, such as computer graphics and software engineering. I have also enjoyed teaching introductory-level courses and believe I can inspire new students and give them a solid foundation in basic computer science concepts and data structures that will carry them to promising careers.