Teaching Statement
Dee A. B. Weikle

A passion for teaching and education has always been a part of who I am. A similar interest in scientific discovery and invention led me to pursue the profession of electrical/computer engineering, but I have always "taught" in one form or another. I tutored in both high school and college. As an engineer, I was frequently the one chosen to mentor new employees and, as an Applications Engineer at Motorola, I organized and taught training sessions for field engineers and customers. Eventually, I decided to pursue an advanced degree so that I could teach and do research full-time. During this time, I taught high school mathematics to at-risk students in the summer Upward Bound program as well as working as a graduate teaching assistant in university classes. Since attaining my PhD, I have taught a first-year engineering design course as an adjunct professor at the University of Virginia and co-taught an upper division research seminar in my position as a Research Scientist in the UVA Computer Science Department.

There are several characteristics I find important to effective teaching. Here I discuss three I feel are particularly important for teaching at universities today. These include teaching students techniques for life-long learning, making learning an experience beyond the simple acquisition of facts and algorithms via traditional lectures, and making a true connection for the student with the material.

Teaching excellence has always included the ability to inspire students to learn on their own. Today, inspiring curiosity and the ability to learn things on ones own takes on an even more important role because technology is changing so rapidly. By the time our current students have worked a year or two, the introductory computer languages, architectures, and specific software we have taught them will have been significantly updated or changed. This means we need to focus on fundamentals that do not change over time as well as teach them techniques for how to learn and update knowledge for themselves. To this end, I incorporate a historical component to my courses as well as inquiry-based problem solving. For example, in an undergraduate computer architecture course I would include lessons on the emergence of reduced instruction set (RISC) machines and how they took advantage of a simple, regular instruction set and new compiler technology to execute programs faster. The focus would be on the trade-offs between simple and regular general-purpose design versus specialized hardware for a given task. I would then support the students in researching their own example in a current machine, ending with an explanation of how current general purpose processors include graphics extensions because of the increased need for graphics in nearly all applications. This project would give students a detailed view of relevant current processors, while teaching them how to make design trade-offs based on parameters that will not fundamentally change as processors change.

Going beyond lectures to teach students has always made good teachers better. The advent of the internet and its companion technologies has made it critical for "bricks and mortar" universities to both incorporate the best of technology in their courses and to

distinguish themselves from virtual universities. One of the best ways to distinguish ourselves is to go beyond lectures and readings and use techniques to enhance a students learning that they cannot get over the internet. Examples include problem based learning and service learning projects that the instructor participates in alongside the student. The first-year design course I taught at the University of Virginia incorporated a design and build project. This course introduces students to the components of design, including building a final design. I implemented a semester long project by introducing the societal problem of "Syndrome X", a combination of diabetes, heart disease, and obesity. The class researched the problem and brainstormed potential solutions while I facilitated. Separate teams designed and presented the top three solutions to three external panelists, a medical doctor, a practicing engineer, and a member of the engineering faculty. From these designs a final design was selected by the class and built. They built an interactive display in the center of campus highlighting the issues surrounding Syndrome X. Students measured the impact of their display by surveying pedestrians a week before it was set up and a week afterward. This project provided the means for students to see a design evolve from beginning to end and taught them teamwork and leadership skills in addition to technical content. This kind of experience can be incorporated into computer program design courses as well.

A fundamental component of my teaching is attempting to meet students where they are in their understanding and then taking them to the next level, expanding their perspective to one that incorporates the new material completely and accurately. If a teacher does not present material from a student's perspective, it is easy for the student to incorporate the new material in such a way as to maintain an incorrect model of that material. To make sure I understand where students are I schedule individual, mandatory meetings with each student at the beginning and I regularly assess class understanding with both formal and informal methods. The individual meetings at the beginning allow me to understand why each student is taking the class, what they hope to get out of it, and how they intend to approach the material. They also learn where my office is, how to reach me, and what my expectations are for the class. During the semester, in addition to regular exams and homework, I have 2-3 chances for students to give me written anonymous feedback on how the class is progressing including content questions. I then summarize these results for the class and make appropriate changes. My teaching style incorporates informal methods of assessing class understanding as well. For example, I might ask everyone to show fingers of how well they understand a concept I just covered or ask how many would like a different explanation by a show of hands. I make every effort to keep the classroom a welcome and comfortable place so that questions are easy to ask and answer.

I can envision myself teaching courses in introductory programming languages, computer architecture, microprocessor design, digital logic design, or applied statistics, as well as other introductory computer science courses. One of the biggest challenges facing computer scientists today is the existence of new multi-chip computers. I think that it is critical that we teach undergraduate computer scientists concepts in parallelization. By the time they graduate, these multi-chip computers will be ubiquitous. I would be excited to work with others in the department on ensuring these concepts are in current courses or creating a new course that would capture the critical concepts across our discipline.