Teaching Statement

Josiah D. Hester
School of Computing
Clemson University

As scientists and life long learners, we have the privilege to be teachers, passing on what we know and inspiring others. This essential part of academia ensures future generations of computer scientists are equipped with the knowledge they need to become inventors, scientists, and engineers. While at Clemson, I was fortunate enough to design curriculum for, and then teach, a course titled *Embedded Systems Prototyping* (CPSC 4820/6820) offered to students in the School of Computing and Department of Computer Engineering. I have also been a teaching assistant for a Software Development Foundations (CPSC 2150) and guest lecturer for Operating Systems (CPSC 3220). I also take an active role in mentoring undergraduate and graduate members of the PERSIST Lab, where I work. These experiences have helped me develop my own philosophy of teaching, and mentoring.

Teaching Philosophy

My teaching philosophy informs how I instruct classes, and how I mentor graduate students and undergraduate researchers, inside and outside of the research lab. In my time as a teacher and mentor I have found that four concepts form the bedrock of my pedagogical style.

Active Learning: Students master technical skills and concepts only through experience, exploration, and struggle. As an instructor, I create an environment where students can easily recognize and understand concepts. My prototyping class featured workshop days where students would build, take apart, program, and breadboard concepts they learned in class. To foster an environment where students could put concepts into practice, I held lectures in the first part of the week, and workshops in the latter half. I would bring resistor kits, LEDs, microcontrollers, 555 timers, and other assorted devices to the class to help the students. The students would take these days to work together to make interesting things or experiment with concepts. I found that this approach helped students tremendously–it got them to work together and use their hands to make something using concepts learned in class. One of the students stated in the end of year evaluations that "The hands on learning experience… made learning incredibly easy."

Engagement: I believe that by engaging students, being excited, and interacting in a pleasant manner I can increase students ability to understand concepts. I found that by going through exercises in front of the class, and fielding questions, students participated more, and ended up more confident in their abilities. I built breadboard prototypes, or live-coded embedded firmware and narrated what I was doing and why (bugs and all). I also found that casting the concepts I was teaching in the scope of my own research interested students. This was easy for the prototyping class; I was able to bring devices from my work (like Ekho, and Amulet) to class—this served the dual purpose of inspiring them to make cool things, and letting them know that their work could make a difference. One student stated, "[Josiah's] personable nature made the class feel more free to explore interesting topics…"

One-on-One Mentoring: One of the most effective forms of engagement comes from interacting with students one on one, in the lab or in the classroom. My students and lab mates and I have spent many hours team debugging code, brainstorming paper topics, and hacking sensors. These sessions reinforce the idea that I am there to help and positively impacted students understanding of the material. I have found that students are more likely to treat you with respect when they see you personally invest in their success.

Equity and Inclusivity: Computer science classrooms are dominated by straight white males, in numbers and in voice. Homogenous environments can leave underrepresented minorities feeling that they do not belong or are not likely to succeed. Computer science concepts are already hard enough without feeling like an outsider. I experienced this to a degree as a Native Hawaiian[1] in computer science. I strive to be as equitable and inclusive in the classroom as possible. As educators and researchers, it is our responsibility to make our classrooms and laboratories safe spaces for women, minorities, and LGBTQI persons. It is our responsibility to push back at the biases we can recognize and fight for equity. There are a variety of techniques that I try to employ to be inclusive. Simple repeated actions are most effective—like highlighting minority and women achievement in computer science as concepts are introduced, using female pronouns in problems sets and exercises, and ensuring that women and minorities are given the chance to engage with the class and the material through questions and office hours. For work in the laboratory setting, enforcing a policy of mutual respect and encouraging students to voice concerns is absolutely necessary.

My goal as a teacher is to help students engage with the material, with me as an instructor, and with each other. In doing so, I hope to equip the next generation of computer scientists and computer engineers to change the world.

Teaching Interests

As a Professor, I look forward to teaching both introductory and advanced courses. I am comfortable teaching a variety of subjects that intersect with my area including Architecture, Operating Systems, Data Structures, and Languages. Naturally, I am also very interested in developing courses (similar to Embedded Prototyping Systems that I developed at Clemson) more closely related to my research. These classes could span topics such as sensor networks, mobile systems, energy harvesting systems, and pervasive computing in general.

[1] Only 0.4% of CS/CE PhD's were awarded to Native Hawaiians/Pacific Islanders in 2015 according to the Taulbee survey.