Teaching Statement

Darren Strash
Institute of Theoretical Informatics
Karlsruhe Institute of Technology
strash@kit.edu
January 12, 2015

Experience and Readiness

During my Ph.D. program at University of California, Irvine, I served as a teaching assistant in various algorithms and data structures courses for one year. For these sessions, I was given carte blanche: I prepared all of the material that I presented; I created many homemade examples, and I actively sought out other materials to give variety and best illustrate the current topic. During one class in particular, I ran two simultaneous discussion sessions (with nearly 100 students combined) as a teaching assistant. In addition to my experience as a teaching assistant, I have given guest lectures on a variety of topics. For each lecture I supplied my own content, making an effort present the material so that students connect with it. I invest a lot of time and effort into my preparation. For me, teaching is not just about effectively transmitting knowledge; my goal is to inspire students to become lifelong learners, and to impart my passion for computer science to them. My experiences as an instructor have ignited in me a passion for teaching, which I am excited to apply in my own classes.

Given my current expertise, I am prepared to teach beginning to advanced algorithms and data structures (both theory and practice), computational complexity, discrete mathematics, security, programming in C++, C, Java, or Python, software engineering, and operating systems courses. With some further preparation, I could also teach cryptography, game development, distributed computing, algorithms for massive data, and scientific computing.

Teaching Philosophy

My teaching philosophy is that students should be constantly challenged, students should learn by doing, and their work and understanding should extend beyond the classroom. For my teaching, I make every effort to make content clear and to make the presentation as dynamic as possible. This means that I work on the whiteboard a lot. When I do present slides, I don’t read from them; they are used as a guide. My intention is for the class to be one long conversation, including all the spontaneity and back-and-forth that is typical of conversation. I frequently ask questions (using the Socratic method), and give many asides
to help build intuition. Most importantly, I make every effort to present a well-rounded view of the concepts I teach: why we choose to study it, it’s history, and how it impacts our world.

A challenge for everyone Having a dynamic classroom means changing the curriculum to meet the needs of the students. In particular, all students should have the opportunity to be challenged by the difficulty of their tasks. This means I actively watch for students who are excelling, and offer them work that will challenge and motivate them. This also means I offer extra support to those who are clearly struggling by providing office hours and by encouraging group study. When students are inclined toward either programming or research tracks, I offer projects that enhance these skills, so that students are prepared to meet their career goals. Ultimately, my goal is to challenge everyone so that they can achieve their potential, and build the skills that will carry them into the future.

Skills for the future While mastery of curriculum is necessary for students achieve their career goals, there are other skills that are equally important for successful career. In particular, I work to give students the necessary skills for the future: critical thinking, effective communication, the ability to work in groups, and ethics. This means that regardless of the topic, students write papers, prepare and present slides, and ideally work on at least one project in a group setting. To enhance critical thinking skills, I try to establish a clear link between the subject and real-world impact. Our work as computer scientists, and the work of those that have preceded us, has undeniably had an impact on the world, and students should know that their work does not happen in isolation.

Impact outside the classroom Though in-class-only projects are necessary to build up students’ skills, students can lose motivation and feel that they are wasting their time being bogged down in projects that lead nowhere. That is why I plan to offer projects that have the potential to be used in the real world.

For those students who are inclined towards programming careers, there are many opportunities to learn from, and contribute to, open source projects. These are tools that I want to use in the classroom. For software engineering and programming classes, I plan to use open source projects as the basis for at least one project. For software engineering, I will encourage students to evaluate the code design and suggest further design improvements. For programming classes, I will have students pick open source projects, become experts in some (small) areas of the code and implement algorithms, design improvements, and additional features.

For those students on a research track, student projects should ideally include a real research component. Understanding and evaluating previous works are essential for starting research, but they are not research in themselves. Ideally, a student’s work should have the potential to result in a publishable research article. This not only helps students to feel that they haven’t wasted their time in a class (when they could be furthering their own research agenda), but I feel that students should have many early research experiences so they can evaluate if the research track is right for them.