Teaching Statement for Neil Klingensmith

As a grad student at UW-Madison, I’ve had experience teaching by supervising junior graduate students and as a TA for a class. It’s been a great experience, and I’ve learned a few things along the way: (1) it’s important to learn by doing, (2) it helps to find a systematic way to solve problems, (3) if you have a good idea, get it out in the community so it can be used.

I think that the best way to learn new things is by doing them. For that reason, I think it’s valuable for courses to have a project component to help students learn how to apply the ideas they learn about in class. I take that tact as a teacher, and I’ll continue to do so as a faculty member. When I supervise other graduate students at UW-Madison, one of the first things I ask them to do is install the Gentoo Linux operating system on a computer. Working with Gentoo—the Linux distribution in which each component is compiled from scratch—can be frustrating because new installations are prone to error. If all the right software components are not installed (or if they’re installed incorrectly), the system will not run. This exercise forces people to really think about what makes a computer tick—a process that most people (even computer programmers) take for granted. With a little guidance, it can be done fairly painlessly.

When fixing a bug or problem in a project, I think it’s important to learn to identify the cause of the problem and understand why the solution worked before moving on. Many times, it is tempting to blindly move on after accidentally or apparently fixing a bug without really understanding why we fixed the bug. Often, though, the root cause of the problem is unanticipated, and identifying it helps us to get a broader understanding of the system we are working on. When presenting final results of a project, it is often helpful to include a short list of some of the problems, root causes, and solutions we encountered along the way. This helps the audience get a better understanding of the nature of the problem and its intricacies.

If we develop something particularly interesting or useful in the lab, I think it is important to get that development out into the world so it can be used by people. Researchers must do this for themselves—no one else is likely to find a project buried in the archives and sell it to the world. One project—Emonix1—was spun off into a startup, and it now operates independently of the university, selling products based on technology developed in the lab.

As a grad student, I’ve worked on several projects with the UW-Madison community that have seen my work put to practical use, and I would continue to do that as faculty. The projects I’ve worked on have largely been collaborations with undergrads or masters students from various backgrounds with various career/life goals.

Working with a diverse group of people, I’ve learned that tailoring the work to the individual is also very important. I try to understand what motivates the student. Helping students to choose projects that fit their goals is important to get a successful outcome.

The grad students I’ve mentored have gone on to work in a variety of jobs: two at startups and several others at large companies, all doing different kinds of work. One of the junior graduate students I mentored wrote the following bit in the preface to his thesis:

*I would like to thank my research mentor, Neil Klingensmith, without whose guidance I wouldn’t have learnt half as much as I have.*

To get such positive feedback about my role in someone’s education was very rewarding. It’s my goal as faculty member to help other people learn about interesting and exciting things in computer science and to develop new and practical technologies.

The courses I would like to teach are related to my area of expertise in IoT computer hardware and software: a survey course on IoT, architecture courses, signal processing, and IoT software. An IoT survey course that covers historical and recent advances in embedded sensing, networking, and software would be useful for a lot of students who are interested in the area. A computer architecture course that covers design and implementation of CPUs, perhaps with a focus on embedded computing would add some breadth for students interested in both architecture and IoT. I would also be able to teach a signal processing or signal analysis course, either focused on classical methods (Fourier, etc.) or statistical techniques. Finally, a course about IoT software including topics on real time systems, embedded networking, and signal processing techniques would be useful for students interested in IoT.

---

1http://emonix.io