

Tips for Teaching

Harry R. Schwartz

December 30, 2014

I've been teaching computer science off and on for a while now. I was a TA for a few years in grad school, and I spent this summer and fall teaching Rails at Metis.

I've come across a few useful teaching guidelines. Having written them down they seem pretty obvious, but I wanted to collect them in one place so I'd have something to reference.

After you've explained something, don't ask, "Any questions?" Instead ask, "What questions do you have?" This little difference invites students to ask questions and makes it harder for them to just say, "nope."

After issuing a call for questions, pause an *uncomfortably* long time. I usually mentally count to six seconds or so. Most of the questions seem to come in the last second or two of that time.

If you have a whiteboard available, use it excessively. Use it more than the projector, if you have one of those. It's awfully hard to pay attention while watching someone write code on a projector, especially if you're just learning to code yourself. Experienced programmers tend to think of live-coding as being interactive and even a little daring, but for a novice it's mostly just dull.

Since you're using the whiteboard, leverage it! Draw pictures and diagrams and arrows! Draw more diagrams and tables than you think are necessary. Students usually appreciate visual taxonomies and charts as a way of organizing the information that you're giving them verbally. Projectors are fine for demonstrating code, but whiteboards are much better for presenting a conceptual framework.

Ask leading questions, even when they seem very simple. "What's the next step?" "What goes here?" "Why are we seeing this error?" "What'll happen when we run this?" Easy leading questions provide useful review and build confidence, and the participation helps keep students alert and engaged.

Every time you introduce a new abstract concept, try to introduce a concrete example or two immediately afterward.

When I first started teaching I'd introduce a concept and ask the class, "Does that make sense to everybody?" I actually expected that students would say "nope!," which was pretty naive. No one wants to admit that they don't understand something—each student assumes that everyone else just *gets it*. Learn to identify the frowns and furrowed brows that indicate that students have questions that they're unable or unwilling to articulate. In a situation like that I'll usually just point out that everyone looks confused. That observation usually gives a couple students the confidence to voice their questions.

During lab time, students will occasionally trap themselves in a shame loop. They won't ask questions initially, assuming that they'll get it. Some time will pass, they'll have made no progress, and they'll begin to feel too ashamed to ask basic getting-started questions. This feeling intensifies over time until eventually the student just wants to run away and never think about the subject again. It's easy to identify someone in this trap—it generally involves a lot of desperate but determined-looking searching and index-flipping. I've found that the best solution for this is to simply describe it and point out that it happens in the beginning of the lab period. If students seem to be getting into this trap I'll occasionally offer "amnesty" by saying something like, "Is anyone stuck at the beginning? That's the hard part!"