

I consider teaching a very important and enjoyable academic duty. While there are many reasons why teaching is important, the most compelling one is that it is one of the most effective ways to pass on our knowledge, thus ensuring that our field grows and prospers. I enjoy teaching for two reasons. First, there is almost nothing more rewarding than seeing someone's eyes light up as they comprehend a new concept. And second, in the process of teaching a difficult concept to others, we ourselves tend to gain a deeper understanding of this concept.

My first teaching duties were as a teaching assistant in the department of computer science at the University of British Columbia (UBC) and spanned a variety of topics including: introductory programming and algorithms, discrete mathematics, operating systems, and formal languages. Most of my duties involved tutoring a group of students who were pursuing a second degree in computer science, the majority of whom came from nontechnical backgrounds.

In the winter term of 2002, I overhauled and then taught CPSC415, a fourth year advanced operating systems course. In the semester preceding the course, I completely reorganized the course, rewrote all the lectures and created all new assignments. I was responsible for all aspects of the course, including: managing three teaching assistants, setting assignments, midterms, final examinations, and lecturing. I also dealt with a serious case of plagiarism. I received a 4.38 out of 5 rating on the student evaluation and received an honorable mention at the department's teaching awards ceremony. I designed the course to be very challenging and was extremely pleased at how well the class did.

I have also served on the curriculum renewal committee that redesigned the UBC computer science undergraduate program. I contributed to the design of a first year course that introduced students to the basics of discrete mathematics and digital circuit design. The course grounds the abstract concepts in discrete math by applying them to concrete problems in digital design. This not only gives students a better understanding of both areas, but highlights the relationships between them.

At the University of Winnipeg, I have taught a variety of courses including: Programming Fundamentals I (ACS-1904), Computer Architecture and Software Systems (ACS-2906), Data Structures and Algorithms (ACS-2947), Computer Networks (ACS-3911), and Principles of Operating Systems (ACS-3931). The latter course, ACS-3931, I developed myself, including the material, lectures, and assignments. I completely overhauled ACS-3911 and ACS-2906, significantly increasing the depth of the material. For all these courses I wrote my own lectures and created completely new assignments. Typically, I have received above 4.0 out of 5 on student evaluations. Based on comments from these evaluations, students tend to find my courses challenging but they appreciate the amount of learning that they do.

My teaching philosophy is highlighted by three points. First, I believe that students get more from a lecture if they are taking notes, versus having them prior to class, processing the information as the lecture progresses. Therefore, I prefer to use a blackboard rather than overheads or Power-Point. Not only does this allow for greater flexibility, but I can be sure that the majority of students will have sufficient time not only to copy what I wrote on the board, but also to think about it.

Second, I believe that it is crucial that students be provided with the "big picture" and how what they are being taught fits into said context. If students understand the relevancy and importance of a topic, they will likely be more motivated to learn. Furthermore, the big picture allows students to create analogies and relations between what they are learning and what they already know. Thus, they will achieve a better understanding of the material and hopefully a greater appreciation of computer science.

Third, and most important, I do not believe in hand-holding the students. It is my experience that setting firm and high expectations, and carrying through on these expectations motivates students to rise to the occasion, resulting in students working more effectively and therefore learning more. I believe that in a field such as computer science, students learn much better if they are forced to discover the answers on their own, through trial and error, rather than have all their “How do I do this” questions answered.

My own teaching interests span a variety of topics. I am interested in teaching general, theory, and systems courses. These include: foundations of computer science, data structures and algorithms, discrete math, complexity, automata theory, digital design, compilers, operating systems, distributed systems.