

Peter Bui, Teaching Statement

I wish to pursue an academic career because I enjoy sharing my passion and enthusiasm for Computer Science with students. During the last ten years, I have been fortunate enough to teach and interact with students while serving as a peer mentor and a teaching assistant in a variety of courses at the University of Notre Dame. Due to my GAANN teaching fellowship from the Department of Education, I have also had the opportunity to serve as the sole instructor for two high level Computer Science courses. The following is my personal reflection on these experiences and what it means to be a good teacher at a university.

Be Well-Prepared

One of the most important aspects of teaching is preparation. This implies a couple of things. First, the instructor must know the subject very well; it is very difficult to teach with authority when the students detect gaps in your knowledge. Sometimes this requires reviewing or learning new material, which was the case for me when I was a teaching assistant for Computer Architecture and Logic Design. Although I am a primarily software-oriented computer scientist, I approached these hardware courses as an opportunity to reinforce my previous education and was able to contribute new lab software and tutorials. From this experience, I learned that teaching is not only an opportunity to share your knowledge, but to grow and expand it as well.

To be well-prepared also means having your teaching materials in order. Few things are more frustrating for students than having a class with missing documents or buggy software. For this reason, when I taught the Computer Graphics (<http://www.nd.edu/~pbui/teaching/cse40166.f10/>) and Programming Challenges (<http://www.nd.edu/~pbui/teaching/cse40872.f09/>) courses, I set up a website that served as the information hub for the course, and posted the syllabus, schedule, lectures, and assignments online. Additionally, I also tested the programming and lab assignments beforehand and had solutions ready. Doing the assignments myself enabled me to anticipate some of the questions students would have, allowed me to fine-tune the material, and aided in constructing clear and fair grading rubrics. Being well-prepared and organized frees the students to focus on the content of the course, rather than the mechanics.

Engage the Students

Another critical component of successful teaching is to engage the students. This means avoiding lectures composed primarily of Powerpoint slides. Such classes tend to bore the students, who end up tuning out. When I served as a substitute professor for a C/C++ programming course, I opted to use the chalkboard intermixed with code demonstrations on a projector. Instead of having a deluge of information, we slowly unwrapped concepts on the board and the students were free to stop and ask questions. The slower pace of chalkboard lectures allowed the students to digest the information and encouraged them to actively process it.

Of course, not all classes lend themselves to writing on the board. For instance, when I taught Computer Graphics, I used a mixture of board-based lectures and slide presentations since some of the material required a richer medium. In these cases, other tactics for engaging the students are required. One such method I have utilized is to have students answer review questions during class. This compels the students to be current with the material and provides a good basis for discussion. Another method I have employed is to have students present their solutions to assignments. This can often lead to interesting debates over programming techniques and strategy, especially if you allow other students to review the solutions. A technique I have not yet tried, but plan on using in the future, is to have students brainstorm answers in small groups and then have a representative report back to the whole class. All of these methods help foster an open and participatory learning environment, which in turn keeps the students focused and engaged.

Having fun and innovative assignments is another effective means of engaging students. In fact, creating programming assignments is my favorite part of teaching. When I assisted in the Fundamentals of Computing course, I designed a lab that required using a graphical toolkit to program a "battleship" game. Having such a fun goal for a lab motivated the students to tackle the topic of graphical user interfaces, something which previous classes struggled to understand. In the teacher evaluations for the Computer Graphics and Programming Challenges courses, students frequently complimented the assignments I designed and noted how useful they were in helping the students master the concepts covered in class. From my experience as both a student and an instructor, I have found that well-designed assignments not only help students learn the material, but also work in keeping students interested and motivated.

Promote Creativity

Good programming assignments also help in another essential part of teaching, which is to promote student creativity. Personally, I have found that I learn best by doing. The beauty of Computer Science is that it is so easy (and fun) to explore and play around with the ideas you learn about; you just need a computer and some inspiration. When creating programming assignments, I try to ensure that each assignment reinforces a specific concept I am trying to teach, while still allowing the students to explore. For instance, in the Computer Graphics course, I often created assignments that required specific methods to be utilized, but left the details of the graphics to the students. So in an assignment involving modeling a vehicle, students came up with all sorts of entertaining programs with objects such as UFOs, rockets, and even a bear riding a scooter. Allowing the students to personalize their solutions enabled them to develop a deeper understanding of the material.

To further promote creativity, I also like to give students large class projects where I specify a few high level goals and let them determine most of the details. Usually, I have the students work in small groups in order to encourage collaboration and teamwork. In the Computer Graphics course, I had two such projects: create a short video and make a 3D computer game. One of the most rewarding moments in my teaching experience was seeing how the students took the concepts I taught in class and created something amazing. For instance, one group re-created the Star Wars Death Star scene for their short video, while another built a virtual chat world with moving avatars for their game. These larger projects have the pleasant side-effect of developing the student's portfolio of work, while serving as avenues for creative discovery and exploration.

Involving undergraduates in your research is also a great way to encourage students to explore different aspects of Computer Science outside the classroom. When I was an undergraduate, I took advantage of various research opportunities both at Notre Dame and at outside institutions such as the Jet Propulsion Laboratory. Later, as a graduate student and instructor, I mentored students on multiple projects that produced software still used today. Such research opportunities greatly augment the educational experience of students by allowing them to creatively apply the lessons they learned in class to real world problems, and to discover and explore their interests in a structured but open manner.

Be Accessible

Finally, it is important to be accessible to your students and interact with them on a regular basis. At the university level, the interaction and relationship between teacher and student is paramount to an effective education. Because of this, I try to foster communication with my students whenever I can. For instance, when I arrived early to the classroom, I walked around and asked the students about the course and how they were doing on the assignments. This gave me a sense of where they were in their understanding, which then allowed me to adjust my lecture or review misunderstood concepts.

During lecture, I frequently solicited questions and comments from the class. Many students were hesitant to ask questions, so I provided other means of communication. One tactic was to welcome students to come and talk to me during office hours. As an instructor, I found these office visits useful not only for the students, but also for myself, since they provided insight into how to better adapt my teaching to meet the students' needs. Another means of interaction was through email. From my experience, electronic communication was much easier for some students, so I let my class know I was open to answering questions through email. Whenever I received an email from a student, I responded promptly and encouraged them to come visit my office.

A final means of soliciting feedback is to have anonymous questionnaires. In the Computer Graphics course, I had the students fill out a brief course evaluation at both the midpoint and end of the course. Because the questionnaire was anonymous, students were free to be honest and frank about their views of the course. From these answers, I discovered how the students viewed the progress of the class and which teaching techniques worked and which needed adjusting. In the end, the purpose of being accessible and open to communication is to ensure that the needs of the students are being met and that they are learning effectively.

Summary

In the future, whenever I approach teaching a class, I will strive to be well-prepared, engage the students, promote creativity, and be accessible, which I believe are core components to effective teaching. I recognize that teaching is a craft that is honed through years of experience and involves continual practice and improvement. For me, the ultimate goal of teaching Computer Science is to ground the students in the fundamental principles and concepts of the field, and to inspire in them my passion and enthusiasm.