Teaching Statement: Donna Dietz
October 2011

My first experience teaching mathematics was in high school when I volunteered to tutor an advanced elementary school student who loved mathematics and was interested in going beyond her curriculum.

Later, I tutored for my department as an undergraduate and then went to graduate school on teaching assistanceships in two different institutions. Since finishing my Ph.D., I have taught in a variety of places, including a community college, a Catholic college, a Pennsylvania state university, and an ivy league university. The courses I've taught range from remedial mathematics to most senior level undergraduate courses for mathematics majors. I currently teach computer science courses with mathematical emphases at the University of Pennsylvania.

As a first priority, I feel I must match my teaching style and expectations to the needs of the students who are enrolled in my courses. I believe that most of us in this field believe this and practice it well. For my lower level students, I pamper them and give them easily attainable goals. In higher level courses, my goals for the students are more complex and involve more synthesis and deeper levels of comprehension. Again, I feel this attitude and practice puts me in the majority, so I will not belabor this point.

The one aspect of my teaching philosophy and practice which has been most under development since graduate school is my response to the mandate that "technology should be embraced as an essential tool for teaching and learning mathematics" (NCTM/NCATE standard 6). The stated goal, of course, is to get technology into grades K-12, and this mandate trickles up to teacher education programs, causing most collegiate programs to be affected in some way, even those which are not involved in teacher preparation.

I feel that "technology infused into every mathematics classroom" has become an unquestionable mantra which really should be re-opened for honest discussion. I feel that technology's role should be more nuanced than what the rhetoric seems to allow.

A quick web search of this phrase shows that this is a huge priority wherever future teachers are being taught and elsewhere. The general attitude one gleans from browsing the web is that disagreeing with this is like saying "kids don't need exercise". Mathematics teaching which does not adhere to this goal is portrayed as lazy and irresponsible. Any teachers who do not use much technology are characterized as simply not wishing to change and are said to be doing a huge disservice to their students, who will need to use computers in their careers.

My own background is that I've always loved using computers and calculators, for myself. However, after I first used them as a teaching assistant, I became frustrated by what I felt to be a disproportionate amount of instruction time for scant additional student comprehension.

For my higher functioning students, the computers are fun and useful. However, for the majority of the students, more time is spent on learning the specifics of the particular software that on the actual mathematics. Glitches or idiosyncrasies in the software derail the comprehension for many students, creating a need to spend even more instruction time explaining why the glitches occur and why they are not mathematically relevant.

As a classical example, the rendering of vertical asymptotes in many early graphing packages confused many of my students about what asymptotes actually were. Other typical graphing issues, such as what happens to sin(1/x) near the origin, often confused students rather than help them. I once had nearly an entire classroom full of students at Rensselaer attempt to calculate "Pi", by using Maple's built-in "Pi", in conjunction with the arctangent method. They were actually excited when their results were "perfect"! That was a major technology-in-the-classroom backfire! (The lab instructions had been proofread and used by multiple professors, and had been fully discussed in class.) Again, the more highly functioning students gained quite a bit from these discussions, while the majority just became frustrated. Perhaps the most amusing overuse of technology in the mathematics classroom is the use of computer software to simulate basic compass and straight edge constructions.

Besides unintended pedagogical side-effects of technology in the classroom, there is a social paradox which has not received sufficient attention. A digital divide is often created when technology use is expected in K-12. I have tutored at an inner-city charter high school in Philadelphia where there is sufficient funding to obtain graphing calculators. However, the students do not actually have access to these devices! They only have very limited in-class access in specific courses, which is not sufficient for most students. These students even struggle to use basic scientific calculators which are only available to most of them while actually at the tutoring center.

In order for students to understand what the technology is doing, they must first comprehend the underlying mathematics. When technology is used to *explain* the mathematics, this creates an issue with priory. To illustrate this, I recall a classroom exchange I had recently with an advanced undergraduate at Penn. The question was posed, "How does the computer figure out whether the permutation is even or odd?". I replied, "You are the programmer now! There are lots of ways to do this! This computer will solve the problem however *you* tell it to!".

I realized that this is the whole "truth" I wish to pass on to both my programming students and my mathematics students. Computers should be seen as a tool, not as a magical box. The human should always be the one in charge, whether on paper or on the computer. Technology should be used only to the extent which it is appropriately understood. It can certainly accelerate learning, but it should not be used without prerequisite understanding.

I believe that there are areas of mathematics where computers can be used very effectively in the classroom and that there are other areas where computers tend to cause more harm than good. Personally, I would not encourage the use to computers by students in the first semester or two of Calculus. When I teach Calculus, I enjoy using Maple animations as classroom demos and prefer not to ask my students to spend their precious study time struggling with Maple syntax, when they have so many fundamental concepts to absorb that semester. (Of course, this preference is heavily balanced by departmental need. If my department is attempting to "infuse technology" into each mathematics course in order to gain/keep accreditation, or for some other reason, then I gladly participate regardless of my personal preference.)

One area of mathematics which seems to lend itself very well to computer technology is, naturally, Discrete Mathematics. If you have a group of students who are beginning programmers, Python is one language which makes it very fun and easy to experiment with number theory, graph theory, encryption, basic statistics, difference equations, and more. (Python is nice because of its clean syntax as well as the fact that integers can be arbitrarily long. Of course, it is also free.) Since my students are using an actual programming language, I have little concern about them viewing the computer as a black box. I can simply allow or restrict their use of specific packages, thereby pitching the exercises to whatever level of detail is appropriate.

Another great mathematics course which works well with programming is linear algebra. I have found MATLAB to be perfect for my use, but Octave (the free version) could easily be used instead if MATLAB is not available. Numerical Computing is another obviously good fit for programming. In my Numerical Computing course, I allowed the use of any programming language. (Those particular students already knew how to program when they enrolled.)

The benefit of using actual programming within the context of a mathematics course is twofold. First, there is the immediate benefit of learning the mathematics, and second, there is the benefit of acquainting the student with fun and marketable computer experience. Many students start out as mathematics majors but later switch to computer science or decide to double major.

A recurring issue for many mathematics majors is that they graduate with no skills outside of pure mathematics, and they often have little time to take electives. Mathematics courses which include programming exercises in lieu of some of their traditional homework could make a big dent in this problem, both for future teachers as well as for those who wish to get a non-academic job after they graduate.

Math majors who are education majors definitely benefit from learning computer science, as they may be called upon to teach it in the future. Mathematics majors who are not education majors also benefit from this exposure, and their ability to land a job may be affected by their computer programming experience. So, either way, mathematics majors really ought to be given these experiences!

As an example, one of my students was recently offered a permanent job programming in Java, following an internship this past summer, which he received after just taking my first-semester programming course which was split between Python and Java. The same amount of programming skill could easily be acquired by mathematics majors if just two or three of their mathematics electives contained programming components.

In conclusion, I believe that the inclusion of technology in various courses should be done carefully and chosen in context of what actually helps the students to learn. Simply using technology as an end in itself is misguided.

Please see a selection of my course materials at my website:
http://www.cis.upenn.edu/~dietzd/teaching_showcase.html