

Teaching statement

Jason Teutsch
teutsch@math.uni-heidelberg.de

August 23, 2011

I am prepared to teach any introductory-level mathematics or computer science courses, including Calculus, linear algebra, and discrete mathematics. Advanced courses I would especially like to teach include theory of computing, set theory, coding theory, cryptography, probability theory, analysis, and point-set topology. I would also like to investigate with students applied topics, such as natural language processing and data mining. I expect to engage students with summer research projects as circumstances permit.

As an Associate Instructor in the Indiana University Mathematics Department (2001–2004), I:

- taught Basic Algebra and Business Calculus,
- led recitations for Calculus I & II,
- conducted help sessions for Finite Mathematics, and
- wrote online WeBWorK homework problems for students.

As a Teaching Assistant in the University of Chicago Computer Science Department (2005–2006), I:

- assisted/lectured Honors Introduction to Computer Science I (in Scheme), and
- assisted Introduction to Computer Science II (in C).

Since my time in Indiana, I have tutored a couple private students. I am fan of the Python programming language as well as SAGE, an open-source mathematics software based in Python. I would not hesitate to use SAGE as a teaching tool in mathematics courses.

In my experience, students generally learn best when they ask questions, in part because the teacher knows better what points to focus on, but also because the process of formulating questions clarifies one's own thought. Individuals bring a wide variety of questions to the classroom. For various reasons, including differing familiarity with the materials, cultural differences, reservation to share, and time

constraints, one has the opportunity to address only a limited number of questions during a class period. Below I describe some successful techniques I have used to address this issue.

When teaching any level of mathematics, I make particular effort to emphasize the importance of the work at hand and overall goals before, and even while, immersing in the details. I like to motivate problems through historical perspective, through extreme cases, irony, and by familiar applications. Strong motivation leads to relevant questions, and questions lead to discussion. I encourage discussion to whatever extent possible because focused dialog yields efficient learning and effective transfer of ideas.

In most areas of study, practice is crucial for skill development, and computer science and mathematics are no exceptions. I encourage students always to work on problems until they get completely stuck and then come talk to me. Getting unstuck is the best way to avoid getting stuck again on the identical problem again later. I welcome students to visit during office hours, especially when they struggle with homework problems. I have found that, occasionally, the student who initially claims to be “bad at math” will later find themselves achieving at a high level once they catch up on the material. I also offer for students to drop by anytime outside of office hours.

Finally, I am a native English speaker, a characteristic which students generally appreciate when I lecture in my mother tongue.