Technically Speaking: Fostering the Communication Skills of Computer Science and Mathematics Students

Jessen T. Havill
havill@denison.edu

Lewis D. Ludwig
ludwigl@denison.edu

Department of Mathematics and Computer Science
Denison University
Granville, OH 43023-0810

ABSTRACT

The Department of Mathematics and Computer Science at Denison University has introduced a significant new oral communication component early in both majors. The sophomore computer science and mathematics majors meet together each week for a "lab" taught jointly by a computer scientist and a mathematician. There were three goals in this endeavor: (1) to prepare students for the workforce and graduate school by improving their oral communication skills, (2) to nurture future researchers in both fields by exposing them to research early in their undergraduate training, and (3) to increase computer science students' exposure to mathematics. In the following, we establish the need for such a course, describe our approach, how it satisfies our three goals, and additional outcomes.

Categories and Subject Descriptors

K.3 [**Computers and Education**]: Computer and Information Science Education—*Curriculum*

General Terms

Human Factors

Keywords

Oral communication, mathematics

1. INTRODUCTION

Many educators (e.g., [2, 10, 15]) and employers [3] agree that undergraduate computer science students need effective communication skills. Due to the highly technical nature of the discipline, it is especially difficult for computer scientists to convey their work to a general audience, such as the board of directors of a company. For this reason, we also agree with the recommendations of CC 2001 [2] and others that oral communication skills are best taught in the discipline. The following quote from the National Association of Colleges and Employers (NACE) report *Job Outlook 2006* [3] summarizes the need.

> Ironically, communication skills not only top employers' list of most-desired skills, but also their list of the skills most lacking in new college graduates. ... Many employers reported that students have trouble with grammar, can't write, and lack presentation skills. Poor communication skills are often evident in the interview, where students are unable to articulate, as one employer said, "how what they have done relates to/contributes to the position" they are seeking.

In addition, as communication is central to research, the practice of oral communication in computer science can serve to introduce students to and excite them about conducting their own research. It is important to nurture good communication skills in our future researchers, as the following indicates [10].

> It also is important for faculty supervisors to help undergraduates grow as researchers. Part of that supervision should include providing the training and experiences that all undergraduate students need to learn effective communication skills that ultimately will allow them to publish successfully in the scholarly literature or to deliver an appropriate presentation to colleagues.

Many educators (e.g., [5, 12]) also agree that a strong mathematics core is critically important in computer science education. As a joint department, we emphasize mathematical foundations throughout our computer science curriculum. In addition, we have long considered collaboration between mathematics and computer science faculty as one of our greatest assets. However, this collaboration was almost non-existent among our students, unless they were double majors. So we began to look for ways to increase interaction, for the students' mutual benefit.

We have introduced a novel oral communication component to our computer science and mathematics majors that endeavors to address these three goals.

Computer science major

Year	Fall	Spring
1	Introduction to CS, Calculus I	Discrete Mathematics, Intermediate Programming
2	Data Structures and Algorithms I	Data Structures and Algorithms II, Computer Organization
3	Theory of Computation	Elective(s)
4	Operating Systems, Elective	Elective(s)

Mathematics major

Year	Fall	Spring
1	Calculus I	Calculus II
2	Introduction to Proof Techniques, Calculus III	Linear Algebra and Differential Equations, Intro to CS
3	Abstract Algebra I, Complex Analysis or Applied Math	Abstract Algebra II, Elective
4	Real Analysis I, Elective	Real Analysis II, Combinatorics or Operations Research

Figure 1: An overview of the computer science and mathematics majors at Denison. The courses hosting our oral communication component are underlined.

2. FINDING THE RIGHT COURSE

The first challenge was to find an appropriate course in each program to implement our idea. As summarized in Figure 1, our typical computer science major takes programming (including basic data structures), discrete mathematics, and calculus during his or her first year, followed by a year of data structures and algorithm analysis and computer organization. Our typical mathematics major takes two semesters of calculus during his or her first year, followed by proof techniques, a third semester of calculus, linear algebra and differential equations, and (recently added) introductory computer science. Both majors take required advanced courses and electives during their last two years. Computer science electives are typically chosen from Computer Networks, Artificial Intelligence, Robotics, High Performance Computing, Programming Languages, and Software Engineering.

We decided to introduce oral communication into our first semester sophomore courses: *Data Structures and Algorithm Analysis I* for computer science majors and *Introduction to Proof Techniques* for mathematics majors. We believe that this time is ideal; students have committed to their major but they still have plenty of time in their undergraduate careers to apply (and hone) their skills. Moreover, if they get excited about research, there is adequate time to pursue this interest during the next two years.

In order to facilitate interaction between the two majors, the two classes meet together once a week for a two hour "lab". In this weekly combined session, they practice delivering computer science and mathematics talks, and learn two "tools of the trade": LaTeX and Waterloo Maple. The two instructors jointly run the combined session. A student's lab grade counts toward 20% of his or her respective course grade; students do not earn any extra credit for the lab.

Our approach lies somewhere between a single presentation of a research project [14, 16] and an entire course devoted to communication methods [9, 17]. Over the 14 week semester, the students obtain a thorough introduction to technical speaking without adding an additional course to the already full curriculum. Moreover, this approach allows us to do justice to oral communication early in the curriculum when students' schedules are especially tight.

3. OVERVIEW OF THE LAB COMPONENT

The lab consists of five main components.

1. Each student presents three talks on topics in mathematics and/or computer science.

2. Each student talk is peer reviewed by four members of the class as well as by the two instructors.

3. Each student writes a review of each of his or her own talks after viewing a recording of the talk and reading the peer evaluations.

4. Each student critiques five outside talks, such as department or campus-wide lectures, over the course of the semester.

5. Each student is introduced to typesetting technical documents with LaTeX and performing basic tasks in Waterloo Maple.

In what follows, we will describe in more detail each of these components. In Figure 2 we present an overview of how the 14 weeks of the semester are allocated.

3.1 Student Talks

We begin each semester with basic instruction on qualities common to good talks in mathematics and computer science. This instruction is based on previous experience with student presentations and two resources detailing the components of an effective talk [11, 20]. We also show a series of video vignettes that are currently being developed at Denison that demonstrate potential speaker pitfalls. For each skill (e.g., proper timing and segmenting of a talk, preparing for questions, and the use of concrete examples [13]), we show a video of both a poor presentation and an effective one.

By week three the students begin delivering talks. We have each student give three talks — 5 minutes, 7 minutes, and 10 minutes in length — over the course of the semester.

Week(s)	Topic(s)
1	Introduction, talk logistics, and LaTeX
2	Critiquing talks and more LaTeX
3 – 5	Students give a round of 5 minute talks
6	Review first round of talks; prepare for second round; Maple
7 – 9	Students give a round of 7 minute talks
10	Review second round of talks; prepare for the last round
11 – 13	Students give final round of 10 minute talks
14	Wrap-up, awards for best talks

Figure 2: Timeline for the oral communication lab.

In order to stress the importance of timing, each talk must be delivered within a minute above or below the stated talk length. These talks count for 10, 20, and 35 percent of a student's lab grade, respectively. (Maple, LaTeX and critiques of outside talks account for the remaining 35 percent.) These three talk opportunities go beyond the minimum recommendations of CC2001 [2] and encourage significant student improvement.

We evaluate the three talks based on nine competencies, which we outline below. The first eight competencies are taken from *The Competent Speaker Speech Evaluation Form* (CSSEF) [1] developed by the National Communication Association, a national professional organization for the field of oral communication. This assessment tool is discipline independent and has met strenuous psychometric requirements for both inter-rater reliability and convergent validity. Below each competency is a clearer explanation based on our interpretation and application of the competency.

Preparation Skills

1. *Choose and narrow a topic appropriately for the audience and occasion.*

   Know the background and experience of your audience and choose an appealing topic that is suitable for the audience's level. Keep in mind that you are addressing both computer science and mathematics majors.

2. *Communicate the main idea in a manner appropriate for the audience and occasion.*

   Explain things at a level understandable to your audience.

3. *Provide appropriate supporting materials based on the audience and occasion.*

   Use concrete examples, props, pictures, and/or tables to convey new terms and definitions.

4. *Use an organization pattern appropriate to the topic, audience, occasion, and purpose.*

   Motivate your talk with an introduction and be sure to end on time with concluding remarks.

Delivery Skills

5. *Use langauge that is appropriate to the audience, occasion, and purpose.*

   Avoid long technical terms or definitions when possible; use examples instead.

6. *Use vocal variety in rate, pitch, and intensity to heighten and maintain interest.*

   If you show enthusiasm for your work, so will your audience.

7. *Use pronunciation, grammar, and articulation appropriate to the designated audience.*

   Do not mumble or use annoying mannerisms such as "umm."

8. *Use physical behaviors that support the verbal message.*

   Don't stay fixed in one place — move!

Media Presentation Skills

9. *Use visual media devices such as overhead, chalkboard, or presentation software appropriate to the topic, audience, occasion, and purpose.*

   Presentation software is great, but do not use too many bells and whistles; these distract from the talk.

For each of the three talks, we assign points to each of these nine competencies. However, in order to ease students into the semester, we evaluate the first talk only on the first three competencies. We later evaluate the second talk on the first six competencies and the last talk on all nine competencies. In the first two talks, the remaining six and three competencies, respectively, are marked for formative purposes only.

Choosing three unique topics for each of the 25 to 30 students per semester is one of the most challenging parts of the lab. Students are directed to several resources and then choose their topics. However, each student is required to have his or her topic approved by the instructor at least a week in advance. This practice aims to maintain the uniqueness of each topic and also serves to steer students away from inappropriate topics, especially those that are either too simplistic or too involved for a particular talk length. The *Math Fun Facts* web page of Harvey Mudd College [19] and the *Classic Fallacies* web page of the University of Toronto [18] are excellent sources of interesting topics for mathematics students, while Dewdney's *The New Turing Omnibus* [8] and some parts of Bentley's *Programming Pearls* [4] contain good topics for computer science talks. Of course, students might also fashion short talks out of topics that they found interesting in their courses or from various conference presentations.

3.2 Peer Review

Each student also receives peer review. During the delivery, the student speaker is reviewed by four peers, randomly chosen by the instructors beforehand and unknown to the speaker. This serves two important purposes. First, it gives the speaker useful feedback from his or her intended audience. Second, it encourages students in the audience to reflect upon what constitutes a good talk. Peer review is difficult for students; we talk about this in class and encourage honest, but constructive criticism. The peer reviews do not affect a speaker's grade, but a reviewer will lose two points for every critique that is not handed in.

The day after the presentations, the peer reviewers submit their critiques on-line based on the nine core competencies outlined above.

3.3 Self Evaluation

After several years of experience with student presentations and conversations with colleagues in our Communication department, we decided to record each student presentation. Instead of distributing edited videotapes or DVD's to each student every week, which would have been prohibitively time consuming, we digitally record each talk and then stream it on the web for the student to watch. For privacy and to lessen the streams' effect on the campus network, we use a firewall to restrict viewing to our department labs.

Each student is required to critically evaluate his or her own talk, after watching the recording and reading the peer reviews. We believe that students must reflect carefully and honestly about their performance in order to improve. Only the speaker truly knows his or her intentions, and can honestly recognize whether the goals of the talk were accomplished.

3.4 Critiquing Outside Talks

While instructing students in good oral communication is an effective way to introduce them to the qualities of a good talk, we have found that students need to actually experience real talks, both good and bad, to improve their skills. To this end, we have students view and critique five outside talks. These critiques count toward 10 percent of the student's lab grade. We find these critiques are more sincere and genuine as they are not critiquing their peers and friends. These talks may be on any topic, although we encourage the students to attend mathematical or computer science talks. Students can generally fulfill this requirement by attending our biweekly departmental talk series (the so-called Faculty and Student (FaSt) talks).

To introduce students to the field of research, we take them to a regional research conference. Usually, this is the Fall Mathematics Conference at Miami University in Oxford, Ohio. Students find this experience invaluable. The conference usually occurs by the sixth week of classes, so each student has delivered his or her first talk. After students see "real" talks, both good and bad, by students and faculty members from other institutions, the quality of the their talks improved markedly. They are allowed to critique two of these talks as part of their required five outside critiques.

The conference has a main research theme each year in which leaders of a field present plenary topics on pertinent research, thus introducing students to the field of research. Quite often, this conference has a special section devoted to applications of computer science in mathematics which the computer science students enjoy. In the coming year, we may also take students to the Midstates Conference on Undergraduate Research in Computer Science and Mathematics (MCURCSM) in October.

3.5 Tools of the Trades

We introduce students to LaTeX and Waterloo Maple early in the semester. We believe that, like competence in oral communication, these are tools that students (and professionals) in both mathematics and computer science should have in their repertoire. Because we are introducing these tools early in their careers, we are able to encourage plenty of practice to gain proficiency. For example, during the semester, both instructors require students to typeset their homework assignments in LaTeX. The assigned exercises in Maple and LaTeX count toward 35 percent of the students' lab grade.

4. OUTCOMES

We conducted an anonymous online survey of all 46 non-graduated students who took the oral communication lab over the last two years. 34 of the 46 students responded. 32 of the respondents reported that they are now more at ease presenting mathematics or computer science material in front of an audience. For example, one student commented:

> Even after I had given a few talks in different classes and different subjects, understanding exactly what makes a good Math/CS talk was invaluable to my confidence in presenting to all sorts of audiences, faculty included. Knowing how to present is one thing, knowing how to present to a specific audience is quite another thing, and in essence is the ultimate goal of a presentation.

Another student said:

> After watching all sorts of people give talks, especially at FaSt talks and [the conference], I feel confident I could present as well as most of them.

We have noticed that more students are willing to present their work at our series of departmental talks. In addition, over the past three years our students have delivered more than twenty-five presentations at regional or national meetings, a dramatic increase. On the survey, of the 25 respondents who had an opinion, 20 reported being more willing to present a FaSt talk. Of the 23 students who reported having given another talk subsequently (in a class or at a NSF-sponsored Research Experience for Undergraduates (REU), summer internship or conference), 18 thought that their experience in the lab improved their presentation; the other five "didn't know". One student remarked:

> This summer, I had an internship where I presented my work to other interns, and later to supervisors. I felt very confident in my oral communication skills in presenting this technical talk, and I credit it entirely to the course (I'd had no previous experience). Further, I feel that my experience showed, and made me a better speaker. It was evident that not many of the other interns had had this sort of perparation.

Anecdotally, the amount of summer research done by our students has also increased, both locally and at REU's around the country. On the survey, 14 students reported that they are now more likely to pursue a research project (19 students "didn't know").

There has been more interaction between our mathematics and computer science majors since they have shared this common experience. Both sets of students have discovered that their fields have much more in common than they imagined. The computer science students can view their peers capably doing and talking about mathematics, making it appear more accessible and less intimidating. But the mathematics students have also learned something about real computer science, hopefully dispelling their misconceptions of it as a "geek-only" enterprise destined to plant them in front of a computer screen for the rest of their lives. Carter [6] provided evidence that many otherwise capable students choose not to major in computer science for just this reason. In addition, Denning, et al. [7] suggest that cross-disciplinary connections might help to increase student interest in computer science. Of the ten students who identified themselves as computer science majors, seven reported that they learned something about mathematics. Of the 17 students who identified themselves as mathematics majors, 12 reported that they had learned something about computer science in the lab. One such student remarked:

> Various presentations described cool ideas that I hadn't seen before, e.g., binary search, Abstract Data Types, etc.

Later courses in both majors have also benefited. Oral presentations in later courses have improved markedly. For example, in the following semester we have the computer science students complete a semester-long research project. At the end of the semester, they teach a class on their topic. The students are more comfortable and therefore convey their topics more clearly and confidently. We can now also ask students in later classes to typeset their documents. Not only does this provide a professional look, but we found students produce better work when their solutions are typed. Although some students commented on the difficulty of learning LaTeX, 25 survey respondents actually thought that learning LaTeX was useful. One student summed it up nicely:

> This class was very helpful in that it is helping make us more professional — we're using the language used when submitting papers and we're learing how to give talks.

5. ACKNOWLEDGMENTS

We would like to acknowledge the work of our colleagues Todd Feil, R. Matthew Kretchmar, and Matthew Neal who have also contributed to and taught these courses.

6. REFERENCES

[1] *The Competent Speaker Speech Evaluation Form.* National Communication Association, 1993.

[2] *Computing Curricula 2001: Computer Science.* ACM and IEEE-CS, 2002.

[3] *Job Outlook.* National Association of Colleges and Employers (NACE), 2006.

[4] J. Bentley. *Programming Pearls, Second Edition.* ACM Press, 2000.

[5] K. B. Bruce, R. L. S. Drysdale, C. Kelemen, and A. Tucker. Why math? *Communications of the ACM*, 46(9):41–44, September 2003.

[6] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 27–31, 2006.

[7] P. Denning, A. McGettrick, P. Rosenbloom, and L. Snyder. Re-centering computer science (special session). In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 65–66, 2006.

[8] A. K. Dewdney. *The New Turing Omnibus.* W. H. Freeman & Co., 1993.

[9] H. A. Etlinger. A framework in which to teach (technical) communication to computer science majors. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 122–126, 2006.

[10] M. A. Fox and N. Hackerman, editors. *Evaluating and Improving Undergraduate Teaching in Science, Technology, Engineering, and Mathematics.* The National Academies Press, 2003.

[11] J. A. Gallian. Advice on giving a good powerpoint presentation. *Math Horizons*, April 2006.

[12] M. D. LeBlanc and R. Leibowitz. Discrete partnership – a case for a full year of discrete math. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 313–317, 2006.

[13] L. D. Ludwig. Technically speaking.... `http://www.denison.edu/mathcs/techspeaking/`, supported by NSF award DUE-0632804.

[14] D. S. Martin. Evaluating oral presentations. *Journal of Computing Sciences in Colleges*, 20(3):48–53, February 2005.

[15] G. McDonald and M. McDonald. Developing oral communication skills of computer science undergraduates. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 279–282, 1993.

[16] C. Norris and J. Wilkes. Computer systems "conference" for teaching communication skills. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 189–193, 1999.

[17] J. A. Polack-Wahl and K. Anewalt. Learning strategies and undergraduate research. In *Proceedings of the SIGCSE technical symposium on Computer science education*, pages 209–213, 2006.

[18] P. Spencer and J. Chan. Classic fallacies. `http://www.math.toronto.edu/mathnet/falseProofs/fallacies.html`, last accessed 11-18-2006.

[19] F. Su. Mudd math fun facts. `http://www.math.hmc.edu/funfacts/`, last accessed 11-18-2006.

[20] J. Zobel. *Writing for Computer Science.* Springer, 2004.