Using Daily Student Presentations to Address Attitudes and Communication Skills in CS1

Chris Bennett
Department of Mathematics and Computer Science
University of Maine Farmington
Farmington, ME 04398
(207) 778-7114
chris.bennett@maine.edu

Timothy Urness
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
(515) 271-2118
timothy.urness@drake.edu

ABSTRACT

Many CS1 courses lack a breadth in coverage of computing-related topics and do not actively engage in non-programming computer science topics. In addition, many introductory (and advanced) courses fail to help students develop oral communication skills. In this paper, we describe our experience with addressing these issues in CS1 courses at two different institutions through the use of brief, daily student presentations. Not only can this help recruitment and retention, but it helps to develop more well-rounded students. We also describe the results of a survey students take before and after the course to evaluate how participating in the course can affect attitudes and beliefs about computer science.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Human Factors

Keywords

Attitudes, beliefs, teaching evaluation, student presentations, communication skills.

1. INTRODUCTION

The traditional CS1 curriculum focuses largely on algorithms and components of designing software. While this is an integral component to computer science, students often struggle to see the connection between the content of the CS1 course and the technology that they use on a daily basis. In an effort to increase students' awareness that computer science involves much more than software design and to help them develop better oral communication skills, we require that each student research and briefly present at least one computer science application at the beginning of a class period. After the presentation, the student is encouraged to lead a brief discussion about the topic.

There are several motivating factors and goals for these presentations. Motivating this assignment is the lack of student-relevant breadth in the introductory course, a lack of active engagement during lecture time with topics beyond programming, and a lack of opportunities to develop and refine important communication skills. The goals include effecting a change in the attitudes of the students regarding computer science as well as providing opportunities to work on presentation skills in a "low-stakes," student-centric environment. A consequent goal is to help with recruitment and retention for the computer science major by allowing students to discover how computer science relates to things that already interest them. Each of these is discussed more in depth in the following section.

The presentations themselves are brief introductions to computing-related topics selected by the students themselves. They are to research and present information in the manner and medium they deem appropriate, and they lead brief discussions afterward to help relate the matter to course material. The presentations are discussed in more detail in Section 3.

Finally, we conducted a survey across multiple semesters at two institutions to assess any changes in student attitudes about computing related issues. Each survey was administered at the beginning and end of each course, and aggregate results indicate an overall improvement in opinions and attitudes toward the discipline. We discuss our results and the possible impact of the presentations on the changes in students' attitudes in Section 4.

2. MOTIVATION AND GOALS

For any assignment, there should of course be expected outcomes. For each outcome, there is usually one or more motivating factors. In some cases, it is a perceived hole in the course material. In other cases, an improved pedagogical technique may lead to a better understanding of the material. It is a combination of these motivating factors that led us to develop the presentation assignment to accomplish the goals of effecting change in student attitudes and improving oral communication skills.

2.1 Goals

There are three goals of the presentation exercise: to change students' attitudes about computer science, to provide an avenue to develop their communication skills, and to have a positive impact on recruitment and retention in the major.

What students think about a discipline can influence how they approach an introductory course and can affect a student's educational experience. Researchers have studied the impact of beliefs and attitudes, and the results of surveys indicate that courses may unintentionally shift students' attitudes away from the attitudes that faculty endorse [1, 8]. Therefore, attitudes or

beliefs regarding the usefulness of the discipline, relevance to the "real world," or the individual's level of interest directly impact the student's educational experience [1].

Because computer science applies to a wide range of topics ranging from architecture to artificial intelligence to cell phones to web applications, students can easily be exposed to career alternatives to programming. Though these applications cannot be studied in depth in an introductory course, the principles of computer science discussed in CS1 can be applied to this diverse set of applications. Daily presentations on student-selected topics can bridge the gap of theory and practice in a way that is meaningful to students and addresses a wide range of topics.

The second goal of the presentation is more direct. Well-developed communication skills are highly-desirable traits of a computer science graduate according to the ACM/IEEE curricular recommendations for computer science, and most if not all institutions identify it as an important part of a larger effort in the general education portion of an undergraduate curriculum [5]. By introducing presentations in a "low-stakes," confidence-building manner, it is easier to integrate the development of these skills into future courses as well.

Finally, a goal consequent to changing attitudes and improving communication skills is improving recruitment and retention for the computer science major [12]. This is a difficult outcome to quantify in this context, but, through a daily connection to "real world" computing and engaging class interactions, students seem more attracted to a major that perhaps previously appeared largely remote and programming-oriented.

2.2 Motivating Factors

There are three motivating factors for using student presentations to achieve our goals. First, there is traditionally a lack of motivating, student-centric "real world" material in an introductory course. While there is little time to delve too deeply into the many topics that might motivate a student to take an introductory computer science course, these presentations are inherently motivating as a student-directed vehicle to introduce a broad range of computing-related topics. Students are immediately invested in the assignment because they select topics that interest them [4 , 6, 11]. Due to the diversity of any given classroom population, the presentation topics are inherently diverse as well. Therefore, this assignment addresses the problem of discussing a broad range of computing-related topics. One theory as to the nationwide decrease in students enrolled in computer science courses is that the discipline of computer science is seen as a group of predominantly male "geeks" that lack social skills. Computer science, of course, is much more than this stereotype allows, and these presentations open the floor to a wide variety of topics.

Second, many non-programming discussions in introductory courses rely on topics the instructor thinks are relevant or happen to be in a "real world" section of a book chapter, and they do not actively engage students on their terms. Student-led presentations and discussions actively engage students in material that is personally relevant. While reading about applications of computing fundamentals can be worthwhile for students, such assignments often suffer from overgeneralizations as well as from student passivity. Researching, presenting, and discussing the material with fellow students produces many of the well-documented benefits of active learning [9], and we have found that students engage in a way that they otherwise would not.

Finally, the presentations are meant to help develop oral communication skills. While this is both a goal and a motivation, many introductory courses do very little to improve students' abilities in this important area. Many employers and other professionals have identified communication skills as something that many current graduates lack, and this can be attributed to the few opportunities there are to develop these skills in most curricula [7]. In addition, the self-selection of computing-related topics often provides confidence in the material, which may lead to stronger preparation and long-term improvement in overall self-confidence.

3. PRESENTATIONS

The primary goal of the presentation is to require each student to think about computer science in a way that would not be traditionally covered in an introductory course and to present their findings to the rest of the class. The discussion immediately following the presentation can serve as a way for the class to discuss the topic or application in a way that is relevant to the material studied thus far, to society, or to the computing community. The presentations are almost daily (with exceptions on exam days, for example), and students may present more than once depending on the size of the section. We require the presentations be only a few minutes long, as to not sacrifice a substantial portion of the class's time or course content. The presentations are designed to be informal and constitute a small percentage of the final grade. At one institution, for example, students did two presentations each (allowing for a chance for improvement) that were a total of five percent of the overall grade.

Human Computer Interaction	Parallel Processing
Artificial intelligence	Virtual Reality
Databases	Numerical Analysis
Software Engineering	Wireless Networking
IPhone	CPUs: Intel vs. AMD
Open Source Software	Netflix $1,000,000 prize
Computer Music	Games: checkers and chess
Bioinformatics	OS: Windows v Mac
One Laptop per Child	Human Computer Interaction
Digital Rights Management	Outsourcing
Forensics	Computer Recycling

Figure 1: A partial list of potential presentation topics presented to students during the first day of class.

At the beginning of the course, students are given an explanation of the presentation component, including a list of possible

subjects. Students are asked to bring to the next class a list of three possible presentation topics (see figure 1). It is necessary to collect several topics from each student as there will inevitably be students that want to report on the same subject. Video games, for example, is a very popular first choice for many students. In this case, variations on themes may be found; one student may discuss the architectural differences in the various gaming platforms while someone else might discuss the sociological aspects of massively multi-player online role playing games (MMORPG).

There was consistently a set of students who wanted to be given topics, despite the fact that they could draw from a list of suggestions. While some of this was out of an expected fear of choosing "wrong", other students acknowledged that this was at least in part due to the problems with making the connection between computer science as a discipline and their everyday interactions with technology. We often found it was better to push the students to find a topic that related to them personally through a series of simple questions to help them find manifestations of computing in their hobbies or major fields of study. This addresses the value students often place on control over their environment [10].

Presentations were held during most lectures (exceptions included exams, for example). They were limited to roughly five minutes in length, but this often led to discussions that exceeded the allotted time. Students are expected to be ready to go at the beginning of class. Managing the time is up to to the instructor., but the authors frequently found the extra time worth it when it helped to illustrate a particularly important point. The number of presentations could be adjusted to fit more or less material, but the daily format reinforced importance of making these connections.

There are other anecdotal observations from our experience with this assignment spanning two semesters at two different universities. Students would often spend a great deal of time in preparation for these brief presentations (as evidenced by the quality of the work) because the subject matter was important to them. Whether the topic was digital rights management, outsourcing, or which gaming platform was the best, students seemed to expend more effort because they wanted to share their passion. This often led to lively discussions with the rest of the class that set up the students and the instructor for "real world" tie-ins to lecture material that otherwise would not have existed. By getting the students talking about something they care about, the presentations go beyond a "blueprint or outline" [3].

With some careful coordination of presentations with particular lectures, it was easy to bring topics to life where making the connection without a student presentation would be difficult. For example, a presentation on a comparison between Sony's PlayStation 3, Microsoft's Xbox 360, and Nintendo's Wii provides a great introduction for not only talking about hardware but also human computer interaction. A presentation on Google Earth helps to make studying different ways of storing data more relevant. A presentation on cryptology helps illustrate the concept of algorithm complexity. Finally, a presentation on famous software failures illustrates the importance of thoroughly testing code.

Another benefit is the relatively strong presentation quality. We attribute this to several factors. First, the topics are generally self-selected so that students are immediately more motivated to research and present their viewpoint. Second, the presentation format itself is informal and "low-stakes" so that the typical stress of doing a prolonged presentation and discussion is minimized. In fact, many presenters seemed more at ease in this environment than in other parts of class when asked to present solutions or help another student in a lab setting. Even those students who were obviously uncomfortable presenting information in the typical format immediately became more comfortable and engaged when asked to explain or demonstrate something in more detail [2]. It is for these students perhaps more than others that practice and exposure to presentation situations is most critical.

Some students cited these presentations as helpful both in making the course more interesting overall and helping to understand concepts from the programming portions of the course. A broad array of topics are covered in large part because of the diverse interests of the students in a typical class. While none specifically thanked the instructors for the opportunity to develop their presentations skills, the practice itself is inherently beneficial. By covering relevant topics that tie to the lecture material, we potentially make a positive impact on the impression of the computer science discipline.

There is one constraint on implementing such a presentation assignment in an introductory course. Because each student should do at least one presentation, such an exercise is limited to relatively small sections of the course. For classes over 30 students, an alternative means of actively engaging the students in computing related topics may be necessary, such as group presentations.

4. SURVEY

In order to assess the impact of the course (and in part the presentations) on students' attitudes and beliefs about computer science, we conducted a survey at the beginning and end of each semester. To eliminate a prejudicial bias as much as possible, we ask students to fill out the following survey before the syllabus is handed out on the first day of class. The survey is then repeated at the end of the course. The students are asked to answer honestly and not to put their names on the paper.

For each item on the survey, we use a Lickert scale where students were asked to indicate if they strongly disagreed, disagreed, were uncertain, agreed, or strongly agreed. The questions are as follows:

1. There are few options available to computer science graduates beyond programming and systems administrations
2. Most computer scientists have poor social skills and relate better to machines than other people
3. Careers in computer science typically rate among the lowest in job satisfaction
4. There are fewer and fewer technology and computing related jobs in the last few years, and new computer science students are having a tough time finding work after graduation
5. Outsourcing has led to a large drop in new domestic technology-related jobs
6. Computer science is essentially the same thing as computer programming

7. The study of computer science is tedious

8. The study of computer science is boring

9. I find it difficult to see the relationship between what I know about computer science and most modern technology that I use

10. The reputation of computer science and its students has affected or will affect whether I will pursue a major or minor in computer science

11. There are fewer opportunities for a female in the male-dominated technology industry

12. Theoretical computer science, such as analysis of algorithms, isn't very relevant to the real world

13. The work computer scientists do in the real world requires a lot of creativity

14. Research in computer science often develops really important ideas that have an impact outside of technology

15. Reasoning skills used to understand computer science material can be helpful to me in understanding things in everyday life

16. If you know what you are doing you can leave work to the last minute and still get it done

4.1 Results

For most of these questions, we observed a positive shift in attitudes about the discipline (see figure 2). There are several questions to which the responses are very telling to the success of the presentations (questions 1, 6, 9, and 14). We believe we can directly attribute the positive improvement in question 9 to eschewing a "textbook" approach in favor of the presentations. By combining a discussion of applications with computer science programming, students get a bigger picture of the discipline. That is, students realize there is more to computer science than just programming (what is typically taught in CS1). Similarly, awareness of these different applications promotes favorable responses to question 11 (gender) and question 12 (theory). The fact that the responses improve over the semester can be partly attributed to a greater exposure to the "bigger picture" of computer science.

Other responses are more attributable to the content of the course itself. Questions 3, 7, and 8 reflect on students impressions of computer science. The relative improvement on these questions indicates that students find the discipline more interesting after the class. Question 2 addresses the stereotype of computer scientists as “nerds”. Students' perceptions about this may also be influenced by how they see the instructor or other computer scientists, not only the applications that are presented. Question 13 deals with whether the students see how programming can be a creative endeavor. Finally, question 15 involves using CS in everyday life, and this is where we see one of the most important improvements. We believe this is in large part due to the content presented by the students.

A few responses may be unrelated to any classroom content. Questions 4 and 5 deal with job availability and outsourcing, but this result varied depending on whether presentations were made on the subject (a sign of the potential influence of the presentations). Questions 10 and 16 are difficult to analyze and perhaps would have been better left off of the survey.

5. SUMMARY

Brief, daily students presentations used as an introduction to the day's lecture are a highly effective way to change students’ perceptions about computer science as well as to help them improve their oral communication skills. Most introductory courses rely on canned discussions or textbook exercises on real world technology and do not provide opportunities for refining presentation skills. Therefore, we believe this particular form of student presentation actively engages students in a broad range of computing related topics, positively affecting their attitudes and improving their presentation skills. In addition, there seems to be a positive impact on recruiting and retention to the computer science major.

6. ACKNOWLEDGMENTS

Our thanks to all of the students who participated in the survey and provided feedback about the presentations in the introductory computer science courses at the University of Maine Farmington and Drake University.

7. REFERENCES

[1] Adams, Perkins, Podolefsky, Dubson, Finkelstein, and Wieman, 2006. A new instrument for measuring student beliefs about physics and learning physics: The Colorado Learning Attitudes about Science Survey. *Physical Review Special Topics: Physics Education Research* 2**,** 1, DOI= http://prst-per.aps.org/abstract/PRSTPER/v2/i1/e010101

[2] Angelo and Cross. 1993. *Classroom Assessment* Techniques, 2nd Ed. Jossey-Bass.

[3] Bain. 2004 What the Best College Teachers Do. Harvard Press.

[4] Becker, K. 2006. First principles of CS instruction. *J. Comput. Small Coll.* 22, 2 (Dec. 2006), 77-84.

[5] *Computing Curricula 2001*, the final report by The Joint Task Force of the IEEE Computer Society and the Association of Computing Machinery on Computing Curricula. December 15, 2001. Available online at http://www.sigcse.org/cc2001. Last retrieved July 28th, 2008.

[6] Hamer, J. 2006. Some experiences with the "contributing student approach". In *Proceedings of the 11th Annual SIGCSE Conference on innovation and Technology in Computer Science Education* (Bologna, Italy, June 26 - 28, 2006). ITICSE '06. ACM, New York, NY, 68-72. DOI= http://doi.acm.org/10.1145/1140124.1140145

[7] Havill, J. T. and Ludwig, L. D. 2007. Technically speaking: fostering the communication skills of computer science and mathematics students. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07. ACM, New York, NY, 185-189. DOI= http://doi.acm.org/10.1145/1227310.1227375

[8] Lewis, C. 2007. Attitudes and beliefs about computer science among students and faculty. *SIGCSE Bull.* 39, 2 (Jun. 2007), 37-41. DOI= http://doi.acm.org/10.1145/1272848.1272880

[9] McConnell, J. J. 2006 Active and cooperative learning: final tips and tricks (part IV). *SIGCSE Bull.* 38, 4 (Dec. 2006), 25-28. DOI= http://doi.acm.org/10.1145/1189136.1189162

[10] McKeachie and Svinicki. 2006 Teaching Tips: Strategies, Research, and Theory for College and University Teachers, 12th ed. Houghton Mifflin.

[11] Murtagh, T. P. 2007. Weaving CS into CS1: a doubly depth-first approach. *SIGCSE Bull.* 39, 1 (Mar. 2007), 336-340. DOI= http://doi.acm.org/10.1145/1227504.1227429

[12] Turner, E. H., Albert, E., Turner, R. M., and Latour, L. 2007. Retaining majors through the introductory sequence. *SIGCSE Bull.* 39, 1 (Mar. 2007), 24-28. DOI= http://doi.acm.org/10.1145/1227504.1227321

Question	End of Semester Average	Beginning of Semester Average	Difference	Result (+/-)
1	1.85	2.18	-0.33	+
2	1.96	2.19	-0.24	-
3	1.71	2.09	-0.38	+
4	1.57	1.73	-0.16	+
5	2.25	2.89	-0.65	+
6	2.29	2.45	-0.17	+
7	2.66	3.01	-0.35	+
8	1.79	2.25	-0.46	+
9	1.54	2.23	-0.69	+
10	2.43	2.28	0.14	**?**
11	1.35	2.23	-0.88	+
12	1.61	2.17	-0.56	+
13	4.31	3.83	0.48	+
14	4.37	3.92	0.45	+
15	4.33	3.90	0.43	+
16	2.41	2.36	0.05	**?**

Figure 2: Data recorded from survey questions listed in section 4. The surveys were conducted at the beginning and the end of the course, and were conducted at two institutions over two semesters with a total of 110 students participating. The rightmost column indicates whether the difference in the response matched with the authors' goals. The responses were scored as follows: strongly disagree = 1; disagree = 2; uncertain = 3; agree = 4; strongly agree = 5.