

Encouraging CS Novices to Write

K. Becker

*University of Calgary
2500 University Drive NW
Calgary, Alberta T2N 1N4
beckerk@ucalgary.ca
403-220-5769
(fax) 403-284-4707*

Abstract

Communication skills are arguably among the most important skills for Computer Science (CS) professionals, and if not *at* the very top of the list, then certainly near it [1]. Yet oddly, other than writing code and associated program documents, CS students are rarely given writing tasks in their CS courses. This paper will examine some possible reasons for why there are few writing tasks other than programming and documentation assigned in CS courses, and what benefits could be realized through more and varied forms of writing. A strategy that has been implemented twice at the author's institution is outlined. It involves the use of short, 250-500 word "reading responses", where students are encouraged to produce what are essentially editorial pieces on some topic relating to the course content. The details of the assignment, as well as the intended outcomes will be outlined. The concept proposed in this paper was implemented in the fall of 2006 in a small class of 30 first year students, and was used again in a larger section of the same course, where there were approximately 100 students. Some initial student reactions are outlined. Suggestions for further examination and development of the concept are suggested.

1.0 Introduction

Communication skills are arguably among the most important skills for Computer Science (CS) professionals, and if not *at* the very top of the list, then certainly near it [1]. Yet oddly, other than writing code and associated program documents, CS students are rarely given writing tasks in their CS courses. The notion of the importance of writing is not new – the "Writing Across the Curriculum" movement was quite popular in the 70's [2], and experienced somewhat of a resurgence in the form of "Writing to Learn" in the 90's [3]. However, somewhere along the way many CS programs seem to have lost touch with this notion. Perhaps this is not surprising – in the quarter century since the initiative first became popular the discipline of computer science has undergone many changes, far more than most other disciplines in the sciences. Similarly, CS Education has changed dramatically in that time. The ACM recommendations for undergraduate programs in 1978 was a 20-page document [4] and outlined what was, at the time, considered to be a comprehensive and fairly complete degree. By contrast, the current document has been divided into four separate volumes, and the CS volume alone totals 240 pages [1].

The discipline of computer science has grown – there is so much "more" to it than there was twenty-five years ago, and yet an undergraduate degree is still a four year

program. One of the consequences of this growth is that many CS faculty feel hard-pressed to cover as much content as possible in the hopes of providing a comprehensive education. For many, this results in a curriculum that ends up being a mile wide but only an inch deep. The focus in CS Education must change from providing students with encyclopedic knowledge of several programming languages and a dozen or so applications and tools, to a thorough grounding in the fundamental principles of CS and skills in problem solving, critical analysis, communication, and other forms of higher order thinking.

Modes and ease of access to factual information have also changed dramatically in the last decade. Ten years ago, this author often told her students that if they wanted to conduct “real research”, they would have to learn how to use the library. Ten years ago, the web was still relatively young and populated largely by commercial sites and general information – much like a glossy or supermarket magazine. Now, it is possible to do the bulk of one’s research from home, while still in your pajamas. About the only sources that still require a physical presence in a library are monographs – and even many of these are now available in digital form online (on sites like book24x7.com).

It is becoming obvious that simply knowing a great number of facts no longer passes for a comprehensive education. On the other hand, it is also true that just knowing where to find a great many facts is also insufficient. “*Knowledge exists only as a function of living tissue*. Knowing where to find knowledge or poems or speeches is nothing like having that material as a part of one’s living tissue. It affects how we think and feel, and education is about precisely these things. The emphasis that has lead away from rote learning, and in this way eventually learning by heart, has been one that gradually and greatly impoverishes minds.” [5 p.68] If we accept this premise and combine that with the notion that the CS body of knowledge has become too large for a four-year program, then the problem becomes a matter of deciding *which* facts and other writings are important to have “as part of one’s living tissue”, and which ones can safely be left “out there” to be Googled as necessary. Perhaps part of the answer comes from allowing at least some of this to sort itself out naturally and organically. This is where writing to learn comes back into the picture. The process of writing one’s thoughts and opinions naturally causes us to review and eventually remember certain facts, passages, events, and people.

2.0 Background

It is known that in order to be effective, lasting, and transferable, that practice must be placed within a meaningful context, and combined with critical feedback that affords the learner an opportunity to re-engage with the material and practice what was learned [6]. Another way to put this is that knowledge must be used and applied in order to be useful, and one way to do this is through writing. Computer Science programs have always included a substantial amount of programming, and this certainly qualifies as use and application of learning. However, this chiefly provides only one perspective, and in a classroom situation, programming is often viewed as a “one-shot” operation: the program is written, tested, documented and submitted for grading; after which that particular solution is rarely looked at again. There are a number of ways of altering this approach

that can make it more closely approximate what students will experience in professional practice [7-9], but that is not the focus of this paper. This paper examines alternative forms of writing not commonly used in CS classes.

These days, there are many more forms of writing than those that existed when the notion of “Writing across the Curriculum” was first conceived of – including, but not limited to: peer-reviewed published works, term papers, research reports, editorial critiques, blogs, online forum postings, chat exchanges and email. Some of these have the potential to improve learning in measurable and significant ways. Email, for example, does not have the same effect as posting to public forums. One reason appears to be that email is seen as “private” and not for public consumption [10], and so tends to be treated more casually. This is important because there is a positive correlation between some forms of student use of in class communication and final exam grades, but there is no significant difference between final scores of email users and non-users [10]. The amount of effort students put into their writing has a direct impact on their grades. If we assume that final grades are a reliable measure of student learning, then proof-reading and editing written work are important for learning. This in turn implies that we should find ways to promote this kind of activity if we want to help students develop to their fullest potential.

Writing as communication in the classroom has other benefits as well. A recent case study of introductory CS courses at a large university [11] found that classroom climates in CS tend to be more defensive (evaluative, controlling, strategic, superior, and deterministic) as opposed to supportive (descriptive, problem-oriented, spontaneous, empathic, equal, provisional), and the former is known to be a less appealing environment for women and minorities. Writing in the classroom is one way to move towards a more supportive climate, which may also have the added benefit of helping to retain a greater number of women in the discipline.

3.0 Writing to Learn in CS

Writing supports at least three roles in a CS program: 1) writing is a communicative skill important in the discipline; 2) writing is an effective way to learn as it involves the entire brain in all the processes: doing, depicting, and symbolizing [12], and 3) writing as a means of communication in the classroom helps to foster a more supportive and inclusive climate [11].

Big term papers are rare in CS. One reason for this is that assessing written works is a time-consuming task, and no-one wants to mark them. Another is that many CS faculty feel under qualified to assess written works, and prefer to leave what they see as language arts to the English department [13]. Typically in CS courses, when term papers are assigned at all, they are assigned only once in any given course, and are generally due at the end of the term. This means that although students will sometimes receive feedback on their efforts, they often receive it after the end of term, and they will not have an opportunity to reflect on that feedback and re-apply it in a new effort. Further, CS students are famous for their unwillingness to write – more than a few will tell you that they chose a CS program expressly to avoid those pesky term papers and essays so often required in the humanities and social sciences. The bad news is that CS professionals

have at least as great a need to learn how to develop a cogent argument, and an ability to skillfully critique the writing of others as anyone in the so-called softer sciences. Computer Scientists need to learn the discourse of their discipline. What this means is that our students need more practice writing.

It is widely accepted that by and large, writing improves with practice. In order to improve the writing and communication skills of our CS graduates, we need to provide many and varied opportunities for them to practice. The current solution to this need at the author's institution, in use for the last few decades, is to have another faculty offer a course on technical writing that is either required, or highly recommended. While this is certainly a step in the right direction, this approach has several problems, not the least of which is that "generic" technical writing does little, in fact, to prepare future practitioners (in *any* profession) for the many and varied writing tasks they will face in their professional careers. Further, it is a single course offering, typically taught by faculty who are not computer scientists, and offered to students from many disciplines in addition to CS. Although the skills taught are important and are often well-taught, it is not enough. Skills cannot be taught in isolation from the context in which they need to be used, at least, that cannot be the only exposure that students have [14]. All of this implies that if we are to help our CS students learn better communication skills, we must offer them opportunities to practice writing, not only in one course, but in many, and within the context of their own discipline.

4.0 Start Small: Reading Responses

One step towards providing writing opportunities has been added to the second introductory programming course (CS102) taught by the author. This course is intended primarily for CS majors, and its main focus is on learning object oriented programming, currently using Java. There are fairly typical programming assignments in the course that account for 50% of the course grade, as well as two exams, which together account for another 30%. For the remaining 20% of the course grade students are asked to choose from various possible tasks, including doing demos of their programs, writing assessments of the course and its content, writing reading responses, and other tasks that can be arranged through individual agreements with the instructor. Different weightings are given to different tasks. It is the reading responses that are of interest in the current context. While the bulk of the lecture time is taken up with talk of programming in Java, the course outline and schedule suggests various additional readings that are either directly related to the main lecture topics, a peripherally related topic (such as GUIs), or some other item with relevance to CS that has appeared in a newspaper or someplace like SlashDot (/.). These other readings may be related to research, politics, business, health, or any other topic that has an obvious tie to computer science, programming, or professional practice. Students are informed of the readings and given links to the sources, but the topics may or may not be discussed in the classroom. These readings are intended to supplement the more traditional course content and provide some sense of the bigger picture of what it means to be a computer scientist. They are also intended to become possible subjects for students' reading responses. These reading responses require the student to write a short, 250-500 word essay, where students are encouraged to produce what are essentially editorial pieces, as opposed to writing "notes" or a

summary. Students may also choose to write on some other topic relating to the course content.

This approach was implemented in the fall of 2006 in a small class of 30 first year students, and was used again the next semester in a larger section of the same course, where there are approximately 100 students. The rationale for implementing this particular approach was that CS students are often reluctant to write, yet it is important that they practice. One way to get reluctant students to write is to make it easy for them to try by starting small. The size of the item they must produce and the time commitment needed to complete it is small (250-500 words), and the impact this item has on their final grade is also small (3%). As a result the perceived risk to the student is reduced, making it easier for them to try.

Most students were quite tentative with their writing at first, and since there is no set deadline for these responses, it typically took a month or so of classes before the first ones were submitted. Allowing this lead time seems to be significant as the nature and general culture of any particular class will take some time to develop and become known. Generally, the responses that were submitted early in the term tended to be non-committal. They either echoed some opinion offered by the instructor in class, or accepted whatever opinion was offered by the author of the piece being critiqued. Interestingly, this changed once the class had had a chance to develop as a community. The responses submitted then tended to be far riskier in terms of the opinions they offered, in some cases disagreeing entirely with the instructor's bias. In this course teaching assistants typically marked the assignments, but all responses were read and graded by the instructor herself. Each response was read and the instructor added some comments on both the content and writing style. In many cases subsequent submissions were longer and more involved than initial ones, indicating that the first reaction to their writing had had a positive effect on the students.

5.0 Conclusions

Whether this practice had a measurable effect on student grades is not known at this time, but the feedback offered during the term indicated that students liked this approach and generally found that it enriched the course as a whole. They appreciated the connection that this activity created between what they were learning in school and what was happening "out in the world". Writing of this sort fosters a connection and interaction between the instructor and the students on a more personal level than some other activities, partly because the task deals with opinions, rather than just facts. This in turn helps build community and a sense of membership in a community of practice.

The use of current events allows us to essentially kill two birds with the same stone by bringing research and timely events into the classroom, thus helping students to learn about *being* a computer scientist in addition to the typical learning *about* computer science.

The approach described in this paper helps to encourage contacts between students and faculty, uses active learning techniques, and respects diverse talents and

ways of learning. These are three of the “Seven Principles for Good Practice in Undergraduate Education” as defined by Chickering and Gameson in 1987 [15]. It was found the amount of additional time required to grade these reading responses was easily counterbalanced by the responses themselves – many of them turned out to be very enjoyable to read!

On the whole the addition of reading responses to the course content was a positive experience, both for the students and the instructor, and will be continued. There are, of course other ways to encourage writing, such as: letter to an author; book review; writings based on abstracts; writing from a pro- or con- position; write what is already known before a discussion; memos; announcements; hypothetical conversations; etc. [16] Writing in almost all of its forms helps develop critical thinking skills through “reflection in action” [17]. We could all consider including more writing in our computer science courses. Perhaps through further investigation we can even begin to discover whether certain writing tasks are more beneficial to computer scientists than others.

6.0 Further Resources

WAC:

writing is the responsibility of the entire academic community,
writing must be integrated across departmental boundaries,
writing instruction must be continuous during all four years of undergraduate education,
writing promotes learning, and
only by practicing the conventions of an academic discipline will students begin to communicate effectively within that discipline.

WAC Resources:

<http://owl.english.purdue.edu/handouts/WAC/>
<http://owl.english.purdue.edu/handouts/WAC/intial.html>
<http://wac.colostate.edu/intro/>

Writing Guides and Information:

Cleveland State University Writing Center:

<http://www.csuohio.edu/writingcenter/writproc.html>

How To Write a Paper in Scientific Journal Style and Format:

<http://abacus.bates.edu/~ganderso/biology/resources/writing/HTWsections.html>

7.0 References

- [1] "Computing Curricula 2001: Final Report of the Joint ACM/IEEE-CS Task Force on Computer Science Education," IEEE Computer Press, Los Alamitos, CA December 2001.
- [2] S. H. McLeod, "Writing across the Curriculum: The Second Stage, and Beyond," *College Composition and Communication*, vol. 40, pp. 337-43, 1989.
- [3] R. Moore, "Writing to Learn Biology.," *Journal of College Science Teaching*, vol. 23, pp. 289-95, 1994.
- [4] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel, and G. Stokes, "Curriculum '78: recommendations for the undergraduate program in computer

- science— a report of the ACM curriculum committee on computer science," *Communications of the ACM archive*, vol. 22, pp. 147 - 166, 1979.
- [5] K. Egan, *Getting it wrong from the beginning: our progressivist inheritance from Herbert Spencer, John Dewey, and Jean Piaget*. New Haven: Yale University Press, 2002.
 - [6] J. Bransford, National Research Council (U.S.). Committee on Developments in the Science of Learning., and National Research Council (U.S.). Committee on Learning Research and Educational Practice., *How people learn: brain, mind, experience, and school*, Expanded ed. Washington, D.C.: National Academy Press, 2000.
 - [7] K. Becker, "Death to Deadlines: A 21st Century Look at the Use of Deadlines and Late Penalties in Programming Assignments," presented at WCCCE 2006 - The Western Canadian Conference on Computing Education, Calgary, Alberta, 2006.
 - [8] M. A. Patton, "The Importance of Being Flexible with Assignment Deadlines," *Higher Education in Europe*, vol. 25, pp. 417-23, 2000.
 - [9] M. D. Jackson, "A Distance-Education Chemistry Course for Nonmajors," *Journal of Science Education and Technology*, vol. 7, pp. 163 - 170, 1998.
 - [10] M. K. Barbour and M. A. J. Collins, "The Act of Online Writing as an Indicator of Student Performance," presented at 27th, Chicago, IL, October 19-23, 2004, 2004.
 - [11] K. Garvin-Doxas and L. J. Barker, "Communication in Computer Science Classrooms: Understanding Defensive Climates as a Means of Creating Supportive Behaviors," *ACM Journal of Educational Resources in Computing*, vol. 4, pp. 1-18, 2004.
 - [12] J. Emig, "Writing as a Mode of Learning," in *The Writing Teacher's Source Book*, G. Tate and E. P. J. Corbett, Eds., 2nd ed. New York: Oxford U P, 1988, pp. 85-93.
 - [13] J. D. Hartman, "Writing to learn and communicate in a data structures course," presented at the twentieth SIGCSE technical symposium on Computer science education, Louisville, Kentucky, United States, 1989.
 - [14] J. Lave and E. Wenger, *Situated learning: legitimate peripheral participation*. Cambridge [England]; New York: Cambridge University Press, 1991.
 - [15] A. W. Chickering and Z. F. Gamson, "Seven Principles for Good Practice in Undergraduate Education," *AAHE Bulletin*, 1987.
 - [16] B. G. Davis, *Tools for teaching*. San Francisco: Jossey-Bass, 1993.
 - [17] D. A. Schön, *The Reflective practitioner: how professionals think in action*. New York: BasicBooks, 1983.