Improving the Learning Environment in CS I
- Experiences with Communication Strategies

Dorothy Deremer
Department of Mathematics and Computer Science
Montclair State, Upper Montclair, NJ 07043
deremer@apollo.montclair.edu

Introduction

CS I is a crucial experience for undergraduate students. While learning technical skills, students develop their perception of computing as an academic discipline and a profession. The resultant view influences career goals relative to computing and long term attitudes about the computer science enterprise. Consequently creating an environment to improve technical learning and awareness of computer science as a profession and to encourage qualified students to stay with or to consider the discipline as a possible major is a worthy exploration. One model to explore is a learning environment emphasizing minimally obtrusive communication strategies. This model assumes students learn more computer science through the diligent use of communication skills rather than in the typical lecture or laboratory. In this context, a primary pedagogical goal is that every student actively contributes to every class. This goal can be achieved through the communication modes of listening, speaking, and writing.

Listening

That students are effective listeners is a tacit assumption of the lecture or laboratory milieu. Without scrutinizing this premise, the CS I instructor can increase listening with the following actions.

1. Resist using phrases such as "Do you understand?" or "Are there any questions?" In contrast, pose a problem specifically related to the topic under consideration and expect students to respond.

2. Avoid rhetorical questions. Students listen more attentively if they perceive answering questions is their responsibility.

3. Defer to a student for a reaction to a question or a comment from another student. Motivate students to respond to each other rather than to the instructor alone.

4. Request a student summarize the opposing view of another student before giving his or her own view when responses indicate important and serious differences.

5. Refrain from repeating, clarifying, or paraphrasing student answers, questions, or comments. Guide the student in rectifying unsatisfactory responses.

6. Subject women and minority students to the same variety of question difficulty level as white male students.

7. Reduce monologue lecturing to at most 10 minute segments by interjecting communication activities, e.g., asking questions and motivating responses.

8. Supply students with modified copies of the lecture transparencies, notes, and/or worksheets to be filled in during the class.

In summary, be patient with questions, answers, and comments from students. Waiting gives evidence to the expectation and importance of student listening in the vitality of the lecture or laboratory.

Speaking

What students believe they may say influences the tone of the lecture or laboratory. Fostering speaking enhances the involvement of students and informs the instructor about the clarity and level of the presentation. The following unordered actions advance student speaking.

1. Call on students by name who are not participating. Learn the correct pronunciation of the name of every student. Inspire class members to do the same. Do not use phrases such as, "I am bad at learning names."

2. Greet students before each lecture or laboratory and expect them to reciprocate. Contact students who miss more than one class.

3. Select students to return papers and labs to class3.mates before the class begins. Put the grade in a concealed location on the work.

4. Maintain eye contact with students especially with those who are speaking. Move around the classroom.

5. Avoid rewarding students who frequently call out responses when this behavior is not appropriate.

6. Rarely permit one student to interrupt another student even if the interruption is supportive. Allow interrupting only for requesting clarification of the speaker's response.

7. Create scenarios in which students critically analyze and verbally respond to the comments of other students.

8. Resist tolerating foreign, minority, or women students' not speaking. Ask a question. Provide undemeaning guidance in soliciting a response but obtain a response.

9. Refrain from excoriating a student for an unclear question. Give the questioner another opportunity with a phrase such as, "I am not sure I understood your question. Please repeat it in a different way."

10. Support comments, questions, right and wrong answers, and hypotheses as valid responses. Stimulate guessing when no response is forth coming.

As far as possible, speak in a knowledgeable and civil manner. Generally demand that students refrain from arrogant, condescending, or sarcastic speech. Promote speaking which is rational, discerning, inquisitive, and egalitarian.

Writing

Although a neglected strategy in introductory computing courses, writing can be a notable communication tool in CS I. Through writing students organize ideas and consider tentative solutions focusing on what is known and not known. In expressing their responses through writing reticent students and discover they can contribute to a verbal discussion. Writing is also a medium for directing the exuberance of extroverted students. By appropriating the following utilities, the instructor can bring writing into the CS I course.

1. Email and Distribution Lists - Sending announcements, reminders, and congratulatory messages among class members increases informal writing.

2. Spontaneous Writing - Ask each student to write down the answer to a instructor posed question. Follow the writing by discussion of the responses between neighboring students. Allow some students to share their results with the class through the medium of transparencies or chalkboard.

3. Focused Freewriting - In this stream of consciousness writing students write about one topic continuously. For example, students are asked to write continuously for 2 minutes about what they know and/or do not understand about arrays. Topics may also be non-technical such as, "Write continuously for 2 minutes about how you can improve your group's performance." In reviewing the writing, the instructor can learn the students' perception of the topic.

4. Entry and Exit Slips - These slips, literally small pieces of paper, are collected before or at the end of a class. With only a few sentences, entry and exit slips present students' thoughts on one topic or issue. For example, one entry slip assignment is to reflect on which part of a laboratory experience was challenging and why. It is collected as students enter the next class.

5. Software Life Cycle - The requirements specification, analysis, and design are excellent vehicles for students to understand and solve problems and to learn and practice formal writing skills.

6. Narratives - These formal reports are agents for investigating non-technical topics or issues. For example, students can write a narrative about a career in computing. Narratives are also effective in learning to express technical ideas in a report format.

For the most part, provide occasion for writing. Frequently collecting evidence of writing increases the literacy level of students.

Communication Strategies and Groups

Working as a team member is a serious skill with lifetime ramifications. Group activities, the beginning of working as a team, require more instructor intervention and planning. However, they are an excellent channel for supporting the learning of CS I and for practicing listening, speaking, and writing. The instructor can establish advantageous group experiences in CS I by reviewing the following thoughts.

1. Establish the central place of groups in CS I by creating ad hoc groups to solve a simple problem beginning with the first class.

2. Create groups to reflect diversity in ethnicity, gender, age, and academic performance.

3. Foster the development of the group identity. Encourage group members to accept ownership of the group actions.

4. Provide activities for experiencing competition among groups while learning cooperation within groups.

5. Require all group members to exhibit both leadership and followership skills.

Communication Skills in the Literature and in Practice

The "Criteria for Accrediting Programs In Computer Science In The United States" requires that oral and written communication skills be developed and applied in the program under review. These requirements affirm that good communication skills are important not only in the student's undergraduate program but also in his/her career as a computing professional. ACM's The No-Nonsense Guide to Computing Careers claims that non-technical skills, [citing communication skills are some of the most important], are probably a bigger part of a professional's life than programming and design. Computing Curricula 1991 also discusses the important supporting role of communication skills in the undergraduate program. Team projects, independent study, and undergraduate research in computer science are cited as mechanisms in which these skills may be developed. These mechanisms, however, are highly intrusive, not appropriate for all students, and generally only applicable to upper division undergraduates. If communication skills are critical to success then we need to seriously invest in their development beginning with CS I.

This article has presented strategies to address this need not only for majors but also for all students in CS I. The author has used all of the cited strategies over the last three years in her sections of CS I at Montclair State. The strategies were primarily associated with technical topics and secondarily with ethics, professionalism and career options in computing. Montclair State is a large, comprehensive, multipurpose, public, and suburban institution of higher education. The College represents diverse cultural and social backgrounds. This diversity was reflected in one typical section of the course taught by the author. Fifty percent of the 30 students were women. Approximately 30% were of African-American or Hispanic descent, 13% of Asian descent, and 7% of Middle East descent. CS I is the first course in the major but less than 25% of the students initially indicated they were computer science majors. Many students were enrolled because of an interest in computing and/or because the course satisfies a requirement for mathematics and science majors and for the College general education in computer science.

With the support of listening, speaking, and writing strategies, the author attempted to present computing as an academic discipline and a profession. Communication strategies have naturally provided opportunities to motivate majors and to encourage other qualified students to consider majoring in the discipline. These opportunities may be especially significant for women because nationally the percentage of female computer science students appears to be increasing at only a slow rate or even decreasing. Lastly the course environment recognizes students are members of a technological society. Increased understanding of computing through communication strategies should assist all students in making more informed decisions.

Bibliography

Computer Science Accreditation Commission (CSAC) of the Computing Sciences Accreditation Board (CSAB). "Criteria For Accrediting Programs in Computer Science in The United States." New York: CSAB, November 1990.

EQUALS. "Computer Startling Statements." Berkeley, California: Lawrence Hall of Science by the Regents of the University of California, 1986.

EQUALS. "Reminders from Research." Berkeley, California: Lawrence Hall of Science by the Regents of the University of California, 1986.

EQUALS. "Strategies for Motivating Student Interest in Computers." Berkeley, California: Lawrence Hall of Science by the Regents of the University of California, 1986.

Frenkel, Karen A. "Women & Computing." Communications of the ACM 33:11.34-46. New York: ACM, November 1990.

Koffman, Elliot B. Pascal, 4th Edition. Reading, Massachusetts: Addison-Wesley, 1992.

Johnson, David W., Johnson, Roger T. and Smith, Karl A. Cooperative Learning: Increasing College Faculty Instructional Productivity. ASHE-ERIC Higher Education Report No. 4, Washington, D.C.: The George Washington University, School of Education and Human Development, 1991.

Levine, Martin. Effective Problem Solving. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Mark, June. "Beyond Equal Access: Gender Equity in Learning with Computers." Women's Educational Equity Act Publishing Center Digest. June 1992, 2-7. Education Development Center, Inc. Funded by OERI, U.S. Department of Education.

Pearl, Amy, Pollack, Martha E., Thomas, Becky, Wolf, Elizabeth, and Wu, Alice. November 1990. "Becoming a Computer Scientist: A Report by the ACM Committee on The Status of Women in Computing Science." Communications of the ACM 33:11.47-57. New York: ACM, November 1990.

Rettig, Marc, ed. The No-Nonsense Guide to Computing Careers. New York: Association for Computing Machinery, 1992.

Shannon, Kathleen. "Workshop for Critical Thinking: Writing in the Math/ Science/ Computer Science Classroom." Montclair State, Upper Montclair, NJ, May 22, 1991.

*******************************COMMUNICATION Continued On Page 38*****************************

```
Procedure C_Reverse(k:Integer)
Comment: takes puzzle from 0^k to 10^(k-1)
if k = 1 then "rotate(1)"
else begin
    C_Reverse(k - 1)
    "rotate(k)"
    C(k - 1)
    end
end
```

4 Odds and Ends

Once again as in the Towers of Hanoi case there is an iterative algorithm, that produces the same sequence of rotations as the recursive algorithm, but is easier for humans to execute.

Iterative Algorithm: If n is odd (even) then on the odd (even) numbered move rotate spinner 1, and on even (odd) numbered moves make the only other legal move. One reason the iterative algorithm is easy for humans to execute is because of the ease in which the spinners can be put in correct position. If the right part of the housing is raised the bar slides all the way to the left putting spinner 1 over the divot, and if the left part of the housing is raised the bar slides to the right until the first vertical spinner catches in the housing, which is the desired position to use rule 2. Hence, one need only remember, "left, right, left, right, ..." Anyone with a military background should have no problem with this.

While it is perhaps disappointing that there is an iterative bypass, it allows one to perform the demonstration. The natural inclination of a student, after hearing the recursive solution for the first time, is to think that they could now solve the puzzle. However, if asked to actually physically solve SPIN-OUT, almost all students will be unable to emulate the recursive algorithm due to insufficient short term memory to keep track of the recursive call stack. In contrast, it is easy for a student to emulate the iterative algorithm after a quick explanation. Field experiments suggest that the optimal time to solve the puzzle, for $n = 7$, is approximately 60 seconds.

It seems to be good practice to always leave some problems unsolved for the reader. The first is to generalize the recursive algorithm to find an algorithm for reaching one arbitrary configuration from another using the fewest number of rotations. As a second problem, consider the following variant of SPIN-OUT. The divot is moved k spinners widths to the left of where it is currently located. Thus, spinners 1 to $k+1$ may be moved over the divot at any time and, if they are not locked by an adjacent horizontal spinner, may be then rotated. The k spinners to the left of the rightmost horizontal spinner may be moved over the divot and rotated, provided they are not locked by adjacent horizontal spinners. We assume that the bar can be slid arbitrarily far to the left, but may not be removed on the left. The problem is to find the optimal recursive algorithm for going from configuration 1^n to configuration 0^n. We end with a question that the author does not know the answer to, "Is there a nice optimal iterative algorithm for this generalized problem?"

References

[1] A. Dewdney, "Ying and yang: recursion and iteration, the Tower of Hanoi and the Chinese rings", *Scientific American*, bf 251, 19 – 28, November 1984.

[2] S. Minsker. The towers of Antwerpen problem, *Information Processing Letters*, **38**, 107–111, 1991.

[3] J. Wu, and R. Chen, "The towers of Hanoi problem with parallel moves", *Information Processing Letters*, **44**, 241–243, 1992.

*****************************COMMUNICATION Continued From Page 35*****************************

Spertus, Ellen. Why Are There so Few Female Computer Scientists? Cambridge, Massachusetts: proposed MIT Artificial Intelligence Laboratory Technical Report, 1991.

Tannen, Deborah. You Just Don't Understand: Women and Men in Conversation. New York: Ballantine Books, 1991.

Tucker, Allen B., ed. with ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 1991. ACM Press and IEEE Computer Society Press, December 17, 1990.

Vetter, Betty. "Women in Science III: Ferment: yes Progress: maybe Change: slow." Mosaic 23:3.34-41. Washington,D.C.: National Science Foundation, Fall 1992.