yourself, but don't be afraid to admit what you don't know or have little experience with. Stress that you can learn quickly and independently, and give some evidence to back up this claim.

Show enthusiasm and preparation: Indicate that you are really interested in working for this company. Show that you've done your homework and researched a little about this company (since this is a hypothetical company, both the interviewer and interviewee have to engage in a little play-acting here.)

Ask appropriate questions: What is the main activity of this group and how does it contribute to the company as a whole? (Here's where you can show off what you already know.) How many people are in this group and what is their background? What will I be doing in this position? What opportunities are there for additional education or training (especially if the interviewer didn't bring this up)? But, in an initial interview, don't ask about salary. (Although one student asked about the financial stability of the company; perhaps these days they should ask to see the company audit!)

Think before you speak: a company advertised for a job in the Boston area. When the interviewer kindly asked a student why she might like to relocate to Boston, she blurted out "But I don't want to go to Boston!"

Don't let the interviewer get away with anything: Sometimes our interviewer will (intentionally) avoid answering a student's question. Sometimes he/she will (intentionally) push the boundaries of equal opportunity employment practices and ask somewhat inappropriate questions. Students who encounter these situations in the mock interview will be better prepared to deal with them should they ever encounter them later.

Terminate the interview in a definite fashion: I enjoyed speaking with you, when may I expect to hear from you again?

Fellow students and interviewers are great at providing constructive criticism. A feedback form helps them jot notes as the interview progresses so they won't forget things to mention. Another possibility is to videotape the interviews for later review.

Last, have the student write a follow-up letter to the interviewer. While this is somewhat old-fashioned, any tactic that makes the job applicant stand out (in a positive way) should be recommended.

Our students have reported back after real job interviews how helpful this process was. One said "I tried to remember everything I learned in the mock interview, I didn't wiggle, I folded my hands on the desk, I maintained eye contact, I asked questions, and I felt much less nervous than I did for the mock interview." This is just one more instance where a student's "education" is not necessarily confined to the formal class or to the classroom.

Please feel free to email either of us with your comments so we can include them in a future issue. Likewise, for anyone who would like to contribute a guest column, let us hear from you.

Community College Corner

New Computing Curriculum for Two-Year Colleges

Robert Campbell
Department of Mathematics and Computer Science
Grinnell College
Grinnell, Iowa 50112 USA
<walker@cs.grinnell.edu>

The ACM Two-Year College Education Committee is pleased to announce that it has concluded work on its latest curriculum report, titled *Guidelines for Associate-Degree Programs in Computer Science* and now available on the Committee's website <http://www.acm-tyc.org>. This report—the product of eighteen months of work with a wide variety of colleagues—updates and replaces the previous computer science curriculum report issued by the Committee in 1993. It also joins other recent Committee reports to address a breadth of computing programs found in associate-degree granting institutions.

This new report provides guidelines specifically targeted for computer science programs in associate-degree granting institutions. It places a principle focus on programs designed for students intending to transfer into baccalaureate programs, accompanied by deliberate guidelines designed to facilitate matters of articulation. It also presents a body of knowledge for computer science in the two-year college setting, associated learning objectives, and detailed descriptions of computer science and accompanying mathematics courses appropriate for implementation in associate-degree programs.

Three specific implementation approaches for introductory courses provide the backbone for this report: imperative-first, objects-first, and breadth-first; for each, sample courses are detailed, with course descriptions that include a list of topics with the appropriate associated learning objectives. The report also specifies computer science elective courses that have a computer science introductory course sequence as a prerequisite; these electives provide curricula to support a career-oriented graduate as well as content appropriate for transfer consideration. In addition, the report outlines required mathematics courses. It also addresses other aspects requisite to a successful computer science program, including a brief indication of the two-year college environment; administration, faculty, and computing resource issues; and requirements from other disciplines.

The *Guidelines* speak to a breadth of issues, including:

- These guidelines call for the incorporating into the computing curriculum a breadth of professional practices. Students are encouraged to work in teams, solve practical problems in course projects, make presentations, confront issues of privacy and ethics, use current technology in laboratories, and attain real-world experience through cooperative education, internships and other practicum activities.
- Communication skills must be identified, developed, nurtured and called upon throughout a computer science degree program. A student must master effective writing, speaking, and listening abilities, and then consistently demonstrate those abilities in a variety of settings, including formal and informal, large group and one-on-one, technical and non-technical, point and counter-point. Such activities should occur across the breadth of the curriculum and in particular, should substantially appear in computer science courses.
- A familiarity with the scientific method (summarized as formulating problem statements and hypothesizing, designing and conducting experiments, observing and collecting data, analyzing and reasoning, and evaluating and concluding) is extremely valuable in the setting of computer science studies.
- The crucial role of mathematics in the foundation of computer science requires that students initiate their mathematics studies early in the pursuit of this degree program. In parallel with their three-course introductory computer science sequence, students must complete a two-semester discrete math sequence (detailed in the report). The learning objectives associated with discrete mathematics specifically support this degree program, and a mathematics department or a computing department (or jointly) should deliver such content with that intent. The discrete mathematics should be an infusion with computer science applications and instructors should continuously reinforce such topics in subsequent computer science courses.

The *Guidelines for Associate-Degree Programs in Computer Science* report shares common goals and outcomes with the recent computer science curricula recommendations for baccalaureate programs developed by the Joint Task Force on Computing Curricula 2001 established by the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) and the Association for Computing Machinery (ACM). The IEEE-CS and ACM recognized the need to revise and update their previous curricular reports; the new *CC2001 CS report* can be found at <http://www.computer.org>.

While the *Guidelines for Associate-Degree Programs in Computer Science* report can be used and implemented independent of the baccalaureate report, these two computer science curriculum reports are specifically designed to complement each other. The two-year college report promotes articulation by enabling computer science faculty in two-year colleges and universities to compare curricula on a topical basis using this report together with the new IEEE-CS/ACM CC2001 Computer Science Report. The two reports share structural elements and content where appropriate. Furthermore, each of the three-course sequences in each of the three approaches in the two-year college report is fundamentally equivalent to the corresponding sequences in the IEEE-CS/ACM *CC2001 CS Report*, and each is designed to be at least equivalent to the corresponding two-course sequence in the *CC2001 CS Report*. Students completing any of the three-course sequences described in the two-year college report should be well prepared for further work in computer science.

The ACM Two-Year College Committee expresses its profound appreciation to everyone that assisted in and supported this undertaking, and we invite the community of computer science educators to read the *Guidelines for Associate-Degree Programs in Computer Science* report. We look forward to hearing from you!