DOCUMENT RESUME

ED 392 417 IR 017 708

AUTHOR Parrish, Allen; And Others

TITLE Assessing Computer Knowledge among College Students.

PUB DATE 95

NOTE 10p.; In: "Emerging Technologies, Lifelong Learning, NECC '95"; see IR 017 705.

PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.

DESCRIPTORS *College Students; *Comparative Testing; Computer Literacy; Computer Science; *Computer Science Education; Curriculum Development; Familiarity; Higher Education; *Knowledge Level; Majors (Students)

IDENTIFIERS Beginning Competence; Breadth of Perspective; *Computer Science Education Research

ABSTRACT

This paper reports on a study involving the administration of two examinations that were designed to evaluate student knowledge in several areas of computing. The tests were given both to computer science majors and to those enrolled in computer science classes from other majors. They sought to discover whether computer science majors demonstrated greater knowledge than non-majors about traditional computer science concepts, persons and events in the history of computing, and general "literacy" with products on the commercial computing market. The tests were also developed to measure the degree to which the introductory computer science course provided high-level familiarity with the breadth of the discipline. Results indicated that computer science majors are substantially more familiar than non-majors with computing topics in general, even those topics not explicitly targeted by the curriculum. Testing also revealed that students just completing a "breadth-first" introductory computer science course did not necessarily exhibit knowledge spanning the entire breadth of computing. Seven sample test questions are provided, along with seven tables displaying test results in each section. The findings of the study may be used to develop a senior capstone course which addresses the areas of computing history, the commercial product market, and legal and ethical computing issues. (Author/BEW)

Assessing Computer Knowledge among College Students

Paper presented at the NECC '95, the Annual National Educational Computing Conference (16th, Baltimore, MD, June 17-19, 1995.

paper

Assessing Computing Knowledge among College Students

Allen Parrish
The University of Alabama
Tuscaloosa, AL 35487-0290
(205) 348-6363
parrish@cs.ua.edu

Susan Vrbsky
The University of Alabama
Tuscaloosa, AL 35487-0290
(205) 348-6363
vrbsky@cs.ua.edu

David Cordes
The University of Alabama
Tuscaloosa, AL 35487-0290
(205) 348-6363
cordes@cs.ua.edu

Crystal Fortner
Lander University
Greenwood, SC 29649
(803) 229-8617

Tracy Camp
The University of Alabama
Tuscaloosa, AL 35487-0290
(205) 348-6363

Gwendolyn Scales
Grambling State University
Ruston, LA 71270
(318) 251-8205

Key words: computer science, outcomes assessment, computer literacy

Abstract

This paper reports on a study involving the administration of an assessment instrument designed to evaluate student knowledge in several areas of computing. The purpose of this study was two-fold. First, we were interested in determining whether our computer science curriculum impacts knowledge of computing topics that are not part of the explicit objectives of courses within the curriculum. This includes areas such as computer literacy (e.g., how to read technical specifications in computer advertisements), the history of computing, and events in commercial computing. Second, we were interested in measuring the degree to which a "breadth-first" CS introductory course sequence provides high-level familiarity with the breadth of the discipline. Our results indicate that computer science majors are substantially more familiar than non-majors with computing topics in general, even those topics that are not targeted explicitly by the computer science curriculum. In addition, students just completing a breadth-first course do not seem to exhibit knowledge spanning the entire breadth of computing.

Paper

Introduction

The topic of outcomes assessment is becoming increasingly important in many areas of education (Light, 1992). Given that computer science sometimes means different things to different people, such assessments can be important tools in determining the concepts that are being conveyed as part of a computer science education. In previous work, we discussed the construction of an outcomes assessment instrument to measure the familiarity of computer science majors with the breadth of the computing discipline (Cordes, *et. al.*, 1994). This assessment instrument took the form of a multiple-choice/matching exam, where questions were directed toward assessing high-level familiarity with computing concepts. We reported the findings of administering this exam in a limited setting, which resulted in several observations with potential of providing guidance for both curriculum design and future research.

Based on the results from our previous work, we administered a second exam to an expanded group of students, including computer science majors at various points within the CS curriculum, as well as non-majors enrolled in CS courses from disciplines such as engineering and management information systems. Thus, we were able to compare the performance among CS majors at various points in the curriculum, as well as between CS majors and "computer literate" non-majors. We constructed this second exam with emphasis in three major areas:

- Traditional computer science concepts. This included coverage of technical concepts that would normally be addressed in computer science courses. Our objective was to select a broad set of basic questions, where the underlying concepts have been covered in a breadth-first introductory sequence.
- Computing history and contributors. This includes both historical events in computing as well as the major contributors to the field. It also included a number of questions from the commercial computing arena, requiring a familiarity with major computing commercial enterprises, as well as with the leaders who contributed to these enterprises.
- Computer literacy. This area included coverage of concepts needed to purchase, install and utilize products available in the commercial computing market. In general, this area included coverage of material that might be found in a computer literacy course for non-majors, as well as material generally related to being literate computer consumer.

Of course, the "computer science concepts" area is explicitly covered by our computer science curriculum. However, the other two areas are not explicitly stressed in most computer science curricula (including our own). Nonetheless, we would ideally expect computer science students to be experts in these areas, even to a greater degree than non-computer science students who are frequent computer users. However, our own experience suggests that CS majors are unfamiliar with computing history, must sometimes struggle to read technical specifications in computer advertisements, and are often not particularly good at installing and using computer systems and software. The inclusion of the latter two categories was an attempt to determine the extent to which these observations are actually correct.

In this paper, we report on a follow-up study on the results of administering the new exam. Our study addresses the following questions:

1. Are CS majors significantly more knowledgeable in the above three areas than non-majors who are frequent computer users? While we would expect CS majors to be more knowledgeable regarding computer science concepts explicitly targeted by the curriculum, it is less clear what to expect regarding computing history and literacy concepts. As stated above, we would hope that CS majors are relative experts in both of these areas as well. However, given that these areas are not explicitly targeted by the curriculum, it is quite plausible that the performance of computer literate non-majors is comparable to the performance of CS majors.
2. How do the CS freshman who have just completed our introductory sequence compare to other groups (particularly the more advanced CS students) with respect to the three areas? Our introductory sequence is organized in "breadth-

first" fashion, where the focus is on the breadth of the discipline (addressing issues such as operating systems, artificial intelligence, multiple programming paradigms, etc.), as opposed to the classical approach of attempting to first achieve depth in software design and development. Breadth-first curriculum design in computer science has been a subject of considerable discussion and controversy (Baldwin, 1990; Locklair, 1991; Motil, 1991; Pratt, 1990). Are students completing a breadth-first course obtaining at least high-level knowledge (sufficient to answer multiple choice questions) spanning the breadth of the discipline?

The paper is organized as follows. Section 2 addresses the details of our previous study (Cordes, *et. al.*, 1994). Section 3 addresses the design of the new exam, as well as the overall design of the accompanying study. Finally, Section 4 addresses our results.

Previous Work

In this section, we review the results of our previous study (Cordes, *et. al.*, 1994) from the Spring 1993 semester. The students involved in this study ranged from second-semester freshman to graduating seniors. An exam was constructed with 100 questions that cover the basic areas of the CS discipline, as well as computing history and basic computer literacy. Questions were multiple choice, true/false, and matching.

The exam was presented to the students (unannounced) during a regularly scheduled class meeting towards the end of the Spring semester. The exam was administered to three different classes: a second-semester freshman-level course, a sophomore/junior-level course, and a course consisting primarily of graduating seniors. The three courses involved were:

- CS 124 : Introduction to Computer Science: This course is a second-semester freshman course. All of the students within the course are familiar with the concepts of programming and software development. The class develops a breadth-first introduction to the discipline of computing as a whole.
- CS 325 : Software Development and Systems: This course is part of the "middle tier" of our CS major. After students complete the first year of the discipline, they must complete four additional CS courses (data structures, assembler, discrete math, and this course) prior to moving on to the upper-level courses in the discipline.
- CS 426 : Introduction to Operating Systems: This is one of four required senior-level courses within the major. All students must take this course prior to graduation. The majority of the students in this course were in their last semester when this exam was administered.

The overall performance on the exam was approximately 68% for seniors in CS 426, 57% for juniors in CS 325, and 45% for freshmen and sophomores in CS 124. The 68% average for seniors was consistent with the average GPA of students in the class (a high C). The average scores indicate, however, that the CS 124 students failed to master a substantial percentage of the material. Given that the CS 124 students are freshmen and sophomores, this appears to be an intuitive result. However, in light of the fact that CS 124 is a breadth-oriented course, one might argue that students should have turned in a better performance on a multiple-choice/matching exam, where depth of knowledge is not needed. We revisit this issue in a more rigorous fashion in this paper.

The exam also revealed a weaknesses in the ability to recognize important contributors to the computing field (both historical and contemporary figures) in all three groups of students. We revisit this issue in this paper as well.

Administration of the Second Exam

A second exam was administered during the Spring 1994 semester. The exam consisted of multiple choice questions covering the three areas mentioned earlier: computer science, history of computing and its major contributors, and basic computer literacy. Sample questions from each area are as follows:

Computer Science

1. ADT
   a. Automated Data Transfer
   b. Average Daily Traffic
   c. Aborted Data Transaction
   d. Abstract Data Type
2. Flip-flops are commonly used to:
   a. prolong the life of a value in memory
   b. implement the basic registers in a machine
   c. store negative values
   d. increment variables by one

History, Contributors and Commercial Computing

1. Grace Murray Hopper
   a. was heavily involved with the development of COBOL
   b. worked on the original ABC computers
   c. worked actively to promote the use of analog register accumulators
   d. founded the National Science Foundation
2. Steve Jobs and Bill Gates
   a. worked together to form the first business data processing corporation
   b. are both committed to supporting IBM through the year 2000

c. are identical twins separated at birth
d. founded Apple and Microsoft

3. The UNIVAC was the first machine to focus on data processing, and was used by:
   a. the Department of Defense for payroll
   b. the US Census bureau
   c. the League of Women Voters to help track voter registration
   d. Sears to manage its catalog mailings

Computer Literacy

1. RGB
   a. Register Generated Bitstream
   b. Regular Generation of Baudrates
   c. Register's Global Byteorder
   d. Red Green Blue
2. font
   a. a style of characters
   b. the size of characters
   c. a character set adopted for the United Nations' universal language
   d. a mixture of graphics and text on the same page

Table 1 illustrates the distribution of questions among the three categories. Some questions were placed in both the computer science and computer literacy categories. Questions in the computer science category covered topics from our breadth-first introductory course, while questions chosen for the computer literacy category covered topics related to reading computer advertisements, installation and usage of hardware and software, and basic terminology that educated computer users would normally be familiar with.

	Category	*Number of Questions*
	Computer Science	32
	History/Contributors/Commercial	17
	Computer Literacy	24
Total		73

Table 1

The exam was given (unannounced) during the last week of the Spring, 1994 semester; students were allowed approximately 75 minutes to complete the exam. The exam was given in five separate courses; in addition to the CS major courses discussed in Section 2 (CS 124, 325 and 426), it was also given in two courses for non-majors:

- CS 412 : C Programming for Non-Majors: This course is designed for seniors and graduate students in disciplines other than computer science. The course is divided between engineering and business majors, and focuses on basic C programming concepts with some discussion of implementing basic data structures and algorithms in C. Students enrolled in this course have taken at least one other programming course (Pascal or FORTRAN).
- CS 466 : Information Systems: This course is designed for juniors and seniors in business, who are majoring in management information systems, or minoring in computer science. The course involves two major components: (a) human factors in software development and (b) metrics, cost estimation and software project management. Students enrolled in this course have generally taken at least two programming courses, and have completed a variety of other computing related courses.

Study Results

The results of the exam were analyzed in terms of the five student groups to which the exam was administered (CS 124, 325, 426, 412 and 466), and in terms of the three categories of questions (computer science, history, and literacy). The mean scores for each student group on each exam section (given in percentages correct for each section) are given in Table 2 below. N is the number of students who took the exam in each course.

Course	*N*	*Literacy*	*History*	*Computer Science*
CS 124 (CS freshmen)	44	47%	15%	34%
CS 325 (CS juniors)	32	63%	25%	65%
CS 426 (CS seniors)	27	71%	35%	72%
CS 412 (non-CS, technical)	46	57%	25%	54%
CS 466 (non-CS, business)	60	58%	18%	49%

Table 2–Overall Summary

We performed an analysis of variance (ANOVA) on the means in each of the question categories in the above table. From these, we were able to determine which student groups differed significantly in performance on each question category. In the subsections below, we separately address the results of these ANOVAs for each category.

Computer Science

Table 3 indicates means, standard deviations and standard errors for the computer science area scores. Unlike the percentage scores in Table 2, these scores are expressed in terms of "raw scores," with a maximum score of 32 (i.e., the number of questions in the computer science area).

Course	*N*	*Mean*	*Std. Dev.*	*Std. Err.*
CS 124 (CS freshmen)	44	10.955	3.815	0.575
CS 325 (CS juniors)	32	20.844	4.608	0.815
CS 426 (CS seniors) 2	7	23.148	3.110	0.598
CS 412 (non-CS, technical)	46	17.348	5.396	0.796
CS 466 (non-CS, business)	60	15.733	4.161	0.537

Table 3–Computer Science Statistics

An ANOVA indicated a significant difference among the above mean scores ($p < .0001$). A post-hoc test is therefore needed to compare specific pairs of groups. Given that sample sizes were both small and varied, we used the (relatively conservative) Bonferroni/Dunn post-hoc test. Table 4 contains the results of this test.

Course Pair	*Mean Diff.*	*Crit. Diff.*	*Significant? (.05 level)*
CS 124, 325	-9.889	2.871	Yes
CS 124, 412	-6.393	2.606	Yes
CS 124, 426	-12.194	3.021	Yes
CS 124, 466	-4.779	2.453	Yes
CS 325, 412	3.496	2.844	Yes
CS 325, 426	-2.304	3.229	No
CS 325, 466	5.110	2.705	Yes
CS 412, 426	-5.800	2.996	Yes
CS 412, 466	1.614	2.422	No
CS 426, 466	7.415	2.864	Yes

Table 4–Computer Science Post-Hoc Test

Based on Table 4, two observations can be made:

1. CS 426 students performed significantly better than all groups except the CS juniors in CS 325.
2. CS 124 students performed significantly worse than all other groups of CS majors, and significantly worse than CS 466 students as well.

Observation 1 indicates that by the end of our computer science curriculum, our CS majors have obtained knowledge regarding computer science concepts that is mostly unavailable in non-major courses. This is an intuitively desirable result, and suggests that the technical content in computer science is not simply "common sense" facts that all computer users know, but is instead a specialized collection of concepts that need formal curricular exposure to learn.

Observation 2, perhaps the more interesting result, involves CS 124 performance. The CS 124 students' performance was significantly inferior to the more advanced CS majors. On the surface, this seems intuitive. However, recall that CS 124 was part of a breadth-oriented introduction to the discipline. We can formalize the idea of "breadth-oriented" vs. "depth-oriented" in terms of Bloom's taxonomy (Bloom, *et. al.*, 1956), which is a well-known way of defining educational objectives in terms of the level of cognitive skills to be obtained. Bloom's taxonomy consists of six levels:

- Knowledge
- Comprehension
- Application
- Analysis/Problem Solving
- Synthesis
- Evaluation

The "knowledge" level simply requires the student to recall specifics of the discipline. In contrast, higher levels require more abstract thinking, synthesis and problem solving. The highest level ("evaluation") requires the student to make judgments about the value of various techniques in the discipline, as well as the fitness of methods for their intended purposes.

One possible goal of a breadth-first computing curriculum might be to attain the "knowledge" or "comprehension" levels of the hierarchy with respect to the entire discipline of computing in the introductory sequence. Subsequent courses would then be targeted toward increasing the cognitive level in the hierarchy. On the other hand, in a depth-first curriculum, one might expect to attain the "analysis/problem solving" level in the introductory sequence, but with respect to a smaller number of concepts. Subsequent courses would then be targeted toward increasing the breadth of knowledge over a wider variety of concepts.

Our results (along with our preliminary results (Cordes, *et. al.*, 1994)) provide some concrete evidence that the breadth-first model may not work as stated. As noted previously (Bloom, *et. al.*, 1956), success in achieving the "knowledge" level can be measured via tests such as multiple choice, matching, etc., since most of what is required is simple recognition of terms and recall of their definitions. Since CS 124 students performed significantly inferior to students farther along in the curriculum on a multiple choice test over basic CS concepts, then we conclude that CS 124 students have not achieved the breadth that they are likely to ultimately obtain, even at the lowest level of Bloom's taxonomy.

Thus, even if a curriculum is nominally breadth-first, the term "breadth-first" is probably a misnomer; in our breadth-first curriculum, students are actually only gaining "some breadth first," followed by a continual evolution of breadth and depth throughout the curriculum. It is likely that there is no such thing as a truly "depth-first" curriculum either; students are not mature enough to initially obtain complete depth in an area, and exposure to a breadth of concepts will allow students to visualize new applications that can be used to obtain additional depth. We believe that it is probably best to abandon terms like "breadth-first" and "depth-first" and seek other models as well as terminology that accurately describes such models; efforts to integrate breadth-first and depth-first curricula have already made some progress in this regard (Shackleford & LeBlanc, 1994).

History, Contributors and Commercial Issues

Table 5 indicates means, standard deviations, and standard errors for the raw scores in the history/commercial area (maximum raw score is 17):

Course	*N*	*Mean*	*Std. Dev.*	*Std. Err.*
CS 124 (CS freshmen)	44	2.636	2.273	0.343
CS 325 (CS juniors)	32	4.219	2.661	0.470
CS 426 (CS seniors)	27	4.130	2.762	0.407
CS 412 (non-CS, technical)	46	5.889	2.806	0.540
CS 466 (non-CS, business)	60	3.017	1.882	0.243

Table 5–History Statistics

An ANOVA indicated a significant difference among the above mean scores ($p < .0001$). We again used the Bonferroni/Dunn post-hoc test to compare group pairs; the results are given in Table 6.

Course Pair	*Mean Diff.*	*Crit. Diff.*	*Significant? (.05 level)*
CS 124, 325	-1.582	1.598	No
CS 124, 412	-1.494	1.451	Yes
CS 124, 426	-3.253	1.682	Yes
CS 124, 466	-0.380	1.365	No
CS 325, 412	0.088	1.583	No
CS 325, 426	-1.670	1.798	No
CS 325, 466	1.202	1.506	No
CS 412, 426	-1.758	1.668	Yes
CS 412, 466	1.114	1.348	No
CS 426, 466	2.872	1.594	Yes

Table 6–History Post-Hoc Test

Again, the CS 426 students performed significantly better than all other groups except the students in CS 325. There are, however, no other significant differences among the data. Since topics in this area are not targeted by any of the courses tested, the absence of a significant difference in most cases is not surprising. The fact that a difference was observed between CS 426 and three groups indicates that by the time CS students are seniors, they are learning at least a portion of this material from somewhere (possibly from in-class discussion where a name or historical incident is mentioned or through peer interactions).

On the other hand, the scores for all groups in this area were the worst of the three sections of the exam by a substantial margin. Table 7 is a breakdown of performance (i.e., percent correct) by topic area of each question, for each group of students.

Question	CS124	CS325	CS412	CS426	CS466
ABC Computer	5	3	7	7	2
DEC	20	38	13	52	15
Dijkstra	2	19	17	33	7
ENIAC	5	3	4	0	2
Hoare	0	38	1	11	2
Grace Hopper	2	6	22	22	3
Jobs andGates	23	29	59	59	60
Knuth	2	13	20	44	2
VonNeumann	9	22	24	48	22
T. J. Watson	9	6	20	4	3
Wozniak	11	19	35	19	22

Table 7–Summary of History Performance

The results in this area reflect no improvement over the results in this area on the previous exam (Cordes, *et. al.*, 1994), indicating a need to include such topics explicitly in our curriculum. We have recently developed a new senior-level course, entitled "Ethical and Societal Issues in Computing," which will include coverage of a variety of historical, ethical, societal, and commercial topics. This course will be offered for the first time during the academic year 1994-95; an outline is given elsewhere (Cordes, *et. al.*, 1994). We plan to administer this exam to students who have completed this course, with the hope of obtaining an improvement in overall performance in this area.

Computer Literacy

Table 8 indicates means, standard deviations and standard errors for the (raw) scores in the computer literacy area (maximum raw score is 24):

Course	*N*	*Mean*	*Std. Dev.*	*Std. Err.*
CS 124 (CS freshmen)	44	11.341	4.513	0.680
CS 325 (CS juniors)	32	15.156	4.719	0.834
CS 426 (CS seniors)	27	13.848	4.269	0.629
CS 412 (non-CS, technical)	46	16.926	3.088	0.594
CS 466 (non-CS, business)	60	13.983	3.496	0.451

Table 8–Computer Literacy Statistics

An ANOVA indicated a significant difference among the above mean scores ($p < .0001$). We again used the Bonferroni/Dunn post-hoc test to compare individual group pairs; the results are given in Table 9.

Course Pair	*Mean Diff*	*Crit. Diff*	*Sinificant? (.05 level)*
CS 124, 325	-3.815	2.674	Yes
CS 124, 412	-2.507	2.427	Yes
CS 124, 426	-5.585	2.813	Yes
CS 124, 466	-2.642	2.284	Yes
CS 325, 412	1.308	2.649	No
CS 325, 426	-1.770	3.007	No
CS 325, 466	1.173	2.519	No
CS 412, 426	-3.078	2.790	Yes
CS 412, 466	-0.136	2.255	No
CS 426, 466	2.943	2.667	Yes

Table 9–Literacy Post-Hoc Test

These results essentially parallel the results of the computer science portion of the test. The CS 124 students performed significantly below many of the other groups, while the CS 426 performed significantly above many of the other groups. In fact, the CS 426 students actually performed significantly above both groups of non-majors. We found these results surprising, given that these questions were deliberately designed to be questions that every literate computer user would know. Since many of the topics in this area are not targeted by the CS curriculum, and the topics are basic computer literacy related, we would have expected everyone to be equally prepared for these questions. Yet the senior CS majors outperformed the non-majors, perhaps confirming that many of these concepts are indeed discussed from time to time in majors classes, or that CS majors are simply more frequent (and motivated) computer users than non-majors. Regardless of the reason, this result may help dispel the notion that CS majors are mostly theorists, and are illiterate with respect to the actual purchase, use and installation of computers.

Additionally, the fact that CS 124 students significantly underperformed the other groups may suggest that college freshmen are perhaps not entering college as computer literate as one might expect. Given the degree to which computers are ubiquitous (in high schools, homes, etc.), one might expect a freshman who has elected to major in computer science to be already an experienced computer user. While the freshmen have probably used computers before, this result at least suggests that freshman are not at the level of many of the other students (majors and non-majors) with respect to understanding terminology associated with basic computer use.

Conclusion

In this paper, we have discussed the results of administering an exam to test knowledge among a variety of student groups with respect to various computing related topics. Our results show that, for this particular exam, senior computer science majors were significantly superior to most other groups on all areas, even with respect to areas that the curriculum was not explicitly designed to cover. In addition, freshman computer science majors who have just completed a "breadth-first" CS introductory sequence tended to underperform relative to the other groups on all areas. Given that the exam was multiple choice and did not require high levels of cognitive understanding, this result was somewhat surprising, suggesting that a breadth-oriented introductory sequence is actually insufficient to deliver the entire breadth of the discipline.

Additionally, students as a whole showed very limited knowledge of computing history, contributors and commercial developments. This result has been partially responsible for motivating the development of a new senior-level, capstone course that will cover many of these issues, as well as ethical and legal issues in computing.

Future work will involve efforts to improve our assessment, with attempts to expand the scope to include more questions related to computing history and commercial developments, as well as computer literacy. In addition, we plan to administer the exam to students at other institutions so as to further generalize these results.

References

Baldwin, D. (1990) Teaching introductory computer science as the science of algorithms,. *Proceedings of the Twenty-First SIGCSE Technical Symposium on Computer Science Education* (pp. 58-62).

Bloom, B., Englehart, M., Furst, E., Hill, W., and Krathwohl, D. (1956) Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: Longmans Green.

Cordes, D., Parrish, A. and Vrbsky, S. (1994) Breadth-oriented outcomes assessment in computer science. *Proceedings of the National Educational Computing Conference* (pp. 298-305).

Light, R. (1992) Explorations with Students and Faculty about Teaching, Learning and Student Life. In *The Harvard Assessment Seminars* (pp. 1-50), Cambridge: Harvard University Press.

Locklair, G.H. (1991) The introductory computer science course. *Proceedings of the Twenty-Second SIGCSE Technical Symposium on Computer Science Education* (pp. 235-239).

Motil, J. (1991) Begin-BIG: An approach to the introductory computing course. *Proceedings of the Twenty-Second SIGCSE Technical Symposium on Computer Science Education* (pp. 226-230).

Pratt, T.W. (1990) Upgrading CS1: An alternative to the proposed COCS survey course. *Proceedings of the Twenty-First SIGCSE Technical Symposium on Computer Science Education* (pp. 68-71).

Shackelford, R. and Leblanc, R. (1994) Integrating "Depth First" and "Breadth First" models of computing curricula. *Proceedings of the 25th SIGCSE Technical Symposium on Computer Science Education* (pp. 6-15).