World Bank Reprint Series: Number 377

Marlaine E. Lockheed and Ellen B. Mandinach

Trends in Educational Computing: Decreasing Interest and the Changing Focus of Instruction

Reprinted with permission from *Educational Researcher*, vol. 15, no. 5 (May 1986), pp. 21–26.

Trends in Educational Computing: Decreasing Interest and the Changing Focus of Instruction

MARLAINE E. LOCKHEED *The World Bank* and ELLEN B. MANDINACH *Educational Testing Service*

ABSTRACT: This article presents a rationale for changing the current emphasis of precollege computer courses. It suggests that as an interim approach, to be used before computers are integrated effectively across the curriculum, computer courses deemphasize BASIC programming skills and instead focus on teaching applications software skills. The research evidence regarding the quality of computer literacy courses is reviewed, and the parallel cognitive and affective consequences of programming and applications software are discussed.

The primary source of evidence regarding student interest in specific undergraduate majors comes from reports on college-bound seniors prepared for the College Board (College Board, 1983b, 1984, 1985). More than 800,000 high school seniors annually complete information regarding their intended college majors, choosing from a list of over 115 fields or specializations. Computer science was first listed as a possible major choice in 1974, with subcategories of systems analysis and data processing added in 1975.

Support for the preparation of this paper was provided by Educational Testing Service.

Marlaine E. Lockheed is a Sociologist in the Education and Training Department, The World Bank, 1818 H Street, NW, Washington, DC 20433. Her specializations are the sociology of education, education technology, and gender in education.

Ellen B. Mandinach is Associate Research Scientist, Educational Testing Service, Princeton, NJ 08541. Her specializations are individual differences and educational uses of microcomputers.

Between 1973 and 1983, interest in computer science rose dramatically. Less than 1% of high school seniors chose computer science as their intended major in 1974 (College Board, 1974), whereas in 1983, more than 10% of high school seniors chose it (College Board, 1983b). In the next two years, however, interest in computer science took a sudden downward turn, with the level of interest in 1985 approximating that of 1981 (see Figure 1). This trend represents a decline of 27%, with women's interest decreasing 35% and men's interest 22%. No other intended major suffered such dramatic shifts during the same period of time.

Although this rise and fall may simply reflect the fortunes of the computer industry in general, another explanation for this decline in interest also may be tenable. Students in the 1970s enrolled in untried and novel computer curricula. By 1985, due to their increased exposure to computers and computer literacy courses at elementary and secondary school levels, students had more information with which to evaluate computer science (Becker, 1983a, 1983c; Lockheed, 1985a).

The increase in students taking precollege computer courses began around 1979. In 1980, 15% of elementary and 50% of secondary schools provided students access to microcomputers in their instructional programs (Goor, 1982); by 1985, 82% of elementary and 93% of secondary schools provided students access to computers (Becker, 1985). The most frequent use of microcomputers was for national, state, and locally mandated computer literacy courses (Boyer, 1983; College Board, 1983a; National Science Board Commission on Precollege Education in Mathematics, Science and Technology, 1983). In general, the mandates provide for required, not elective, computer course participation. It is the contention of the authors that the generally poor quality and restricted curriculum of these courses, which in 1985 were required for many high school seniors, accounts at least in part for the declining interest in computer science at the college level.

Inadequate Computer Literacy Courses

Early computer literacy courses, which stressed learning to program in BASIC, were generally of poor quality in terms of teacher training, course cur-

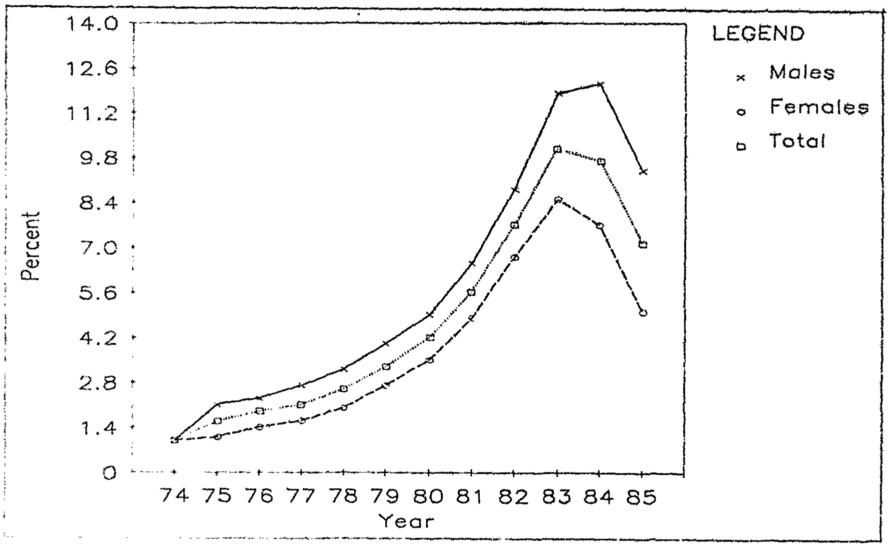

FIGURE 1. *Percent of College Bound Seniors Indicating Primary Interest in a Computer Science Major.*

riculum, time allocated to computer use, ratio of computers to students, and type of hardware and software used. The vast majority of teachers teaching the courses were untrained, with 54.3% of elementary, 59.0% of junior high school, and 58.5% of high school computer literacy teachers rated by their principals as "minimally qualified" to teach the course (Wright, Melmed, & Farris, 1984).

The need to train students in computer programming increased the demand for available computers. Only one-third of elementary school users had 15 minutes (at most) of on-line time per week, and only 2% got more than one hour of hands-on time during a given week (Becker, 1983b). At the secondary level, most students used the microcomputer for 45 minutes a week (Becker, 1983b).

Programming instruction at the precollege level was inadequate in scope, depth, and choice of language. In most cases, students enrolled in college courses had to unlearn misconceptions and poor techniques before they could profit from college instruction (e.g., Linn, 1985). In addition, programming courses apparently had inequitable effects on students; females and minority students gained less access to computers than did males and majority students (Anderson, Welch, & Harris, 1984), and males reported more positive attitudes toward computers than did females (Chen, 1984; Collis, 1984; Fetler, 1985; Lockheed, 1985b; Lockheed, Gulovsen, & Morse, 1985; Lockheed, Nielsen, & Stone, 1983; Wilder, Mackie, & Cooper, 1985).

Declining Interest in Courses

What distinguished the 1985 from the 1983 college-bound seniors was their precollege exposure to computers. Because 1983 college-bound seniors were sophomores in 1981, they were relatively unaffected by the swell of computer literacy courses that were required of junior and senior high school students in that year. In fact, less than 3% of the high school class of 1982 had taken a computer literacy course in school (Rock, Ekstrom, Goertz, & Pollack, 1985).

By comparison, 31.1% of high school juniors reported having completed one semester or more of coursework in computers or programming in 1984 (National Assessment of Educational Progress, NAEP, 1985). Thus, college-bound seniors in 1985 who were eighth-graders in 1981 were quite likely to have been exposed to a computer course before completing high school. We suggest that exposure to hastily developed, inadequately staffed, and poorly supplied computer literacy courses at the precollege level may be a major cause of the observed decline in interest in computer-related college majors and careers.

Long- and Short-Term Alternatives to Programming Courses

Although interest in postsecondary computer-related careers is declining as rapidly as computer literacy courses at the precollege level are increasing, it is possible to reverse this trend by changing the precollege computer course curriculum.

Alternatives to computer programming courses include the long-term objective of integrating computers into the curriculum, and the short-term objective of developing a curriculum based on applications software. Both alternatives will be discussed, and a rationale for emphasizing an applications-based computer literacy course in the short-term will be offered.

Integrating Computers into the Curriculum

One alternative would be to integrate computers into the curriculum as a tool for students (Hunter, 1983; Tucker, 1983). Separate computer courses would not be taught; instead, computer applications would be taught in conjunction with extant curricula. For example, databases on the presidents or the planets could be used as instructional materials in social studies or science courses; word processors could be used in any class that requires writing; graphics packages could be introduced in art, math, or science; and spreadsheets could be used in math, home economics, shop, or physical education. Simulations and games could be integrated into content-appropriate classes. Programming instruction would remain a separate course.

Logistical difficulties argue against this alternative at present. It may be an admirable long-term goal, but as a short-term objective it faces several problems. First, there is a serious lack of curricula developed with computer tools in mind. Recent efforts, such as *My Students Use Computers* (Hunter, 1983) or Scholastic's databases for use with PFS:File represent a small step toward integrating computers into the curriculum, but they should be viewed as preliminary efforts, not full-blown accomplishments.

Second, teachers are inadequately prepared to use computers in the regular classroom (Wright et al., 1984). Teachers other than computer instructors have received, on the average, less than 10 hours of hands-on computer training—little more than what is needed to learn basic keyboard operations. Even training programs that have attempted to assist teachers develop curricula have had to provide introductory "machine handling" skill development, which often uses all the time available for training and leaves little time for curriculum development. Few instructors other than computer teachers currently use computers in their classes, and the vast majority of students use computers primarily in computer classes (Becker, 1985; Lockheed, 1985a; Lockheed, Gulovsen, & Morrison, 1985).

Third, the physical arrangement of hardware in the schools, which often places all computers in a single room rather than distributing them across several classrooms, inhibits continuous widespread use (Becker, 1984). Finally, schools must incur substantial costs both for acquiring and maintaining computers and for retrofitting buildings to accommodate computers.

Applications-Based Course

A second alternative would be to teach an applications-based computer course. The class would include instruction in how to use word processors, database managers, spreadsheets, and graphics packages, and would emphasize generalizable skills related to planning, gathering, and interpreting data. The applications-based computer class could be linked to the remainder of the curriculum by collaborative teaching strategies, and the scope of the class could be enhanced by drawing on materials taken from a number of domains.

Rationale for Choosing Applications

A well-structured computer curriculum based on applications software is preferable to one based on programming for two reasons: it is likely to foster higher cognitive skills similar to those engendered in programming; and it is likely to stimulate positive affective responses (e.g., interest in and liking of computers) for students who heretofore have been disinterested in programming. If applications can produce positive cognitive effects, and at the same time increase affective responses to computer science in a broader range of students, then it seems reasonable to focus computer instruction on such software instead of focusing on programming instruction.

Although little current empirical support for our claim exists at this time, it is possible to suggest why an applications-based curriculum may be generally preferable to one focused on programming.

Parallel cognitive outcomes of applications and programming. A primary reason for selecting an applications-based curriculum is that such software may be likely to foster cognitive skills equivalent to those engendered in programming. This rationale rests on two assumptions. First, learning to program and learning from various computer environments (e.g., applications software, stimulations), more generally, is likely to stimulate the acquisition of higher-order cognitive skills (see Patterson & Smith, 1985). Second, skills identified as important in programming (see Atwood & Ramsey, 1978; Kurland, Mawby, & Cahir, 1984; Linn, 1985; Pea & Kurland, 1984) have analogues in and can be generalized to other computer learning environments. Programs of research (e.g., those at the Bank Street College of Education and the ACCCEL Project at the University of California, Berkeley) are beginning to provide empirical evidence of the cognitive skills engendered in programming. Once these results identify the requisite programming skills, research then can begin to examine the parallels among and generalizability to other computer domains.

Thus, it will be possible to analyze applications software, examine necessary skills, and draw parallels to those required in programming. Although specific skills may not be identical, it is likely that similarities among global constructs in the two computer learning environments can be found. Linn's (1985) chain of cognitive accomplishments for programming provides a general framework for examining the demands and potential outcomes of computer learning environments and can be generalized to an applications-based computer curriculum. Lockheed's research on the skills requisite for database manager use provides preliminary descriptive evidence for these parallels (Lockheed, Gulovsen, & Morse, 1985; Lockheed et al., 1985).

According to Linn's model, there is a three-stage chain of accomplishments for programming: (a) learning language features; (b) learning design skills, including templates and procedural skills; and (c) learning generalizable problem-solving skills such as planning and self-regulation (see Linn, 1985, for details).

The first link in the chain of cognitive accomplishments consists of non-decomposable language features, the elements basic to all formal systems. In a database, menu operations comprise the primatives of the systems, while a "field" or "record" serves as a variable.

The second link in Linn's chain consists of design skills, or techniques for combining language features; design skills include both templates and procedural skills. Although Linn (1985) subsumes procedural skills and templates under design techniques, Mandinach (1985) reasons that the two subcomponents should be distinguished. Templates are distinct from procedural skills insofar as the former combines language features of the system, whereas procedural skills combine language features as well as templates. Common database templates include the sets of commands for conducting a search or for executing a sort. Procedural skills include planning, testing, and reformulating. For example, a plan for developing a database file is outlined, implemented, tested for errors, and reformulated if errors are found.

The third link in the chain of cognitive accomplishments is comprised of templates and procedural skills that can be generalized to other formal systems. For example, recognition of the appropriateness of skills and knowledge for particular situations is a general problem-solving or autonomous learning skill. Autonomous learning skills are those needed to manage one's own mastery of a new formal system and assume responsibility for one's own learning (Corno & Mandinach, 1983; Glaser, 1984). Such skills are high order and logically follow from requisite declarative and procedural knowledge. Thus, in order of difficulty, learning language features is least difficult, followed by templates, procedural skills, and finally generalizable problem-solving skills.

Preferable affective outcomes. Children generally express positive attitudes about computers when they are used for computer-based instruction, game-playing, or Logo turtle graphics (Clements, 1981; Demarest & McKenzie, 1984; Lawton & Gereschner, 1982; Wilder et al., 1985; Williams & Williams, 1984). However, their attitudes about computers used for required instruction in programming languages—the typical use of computers in secondary schools (Becker, 1985)—range from neutral to negative. An example of this finding is provided by a study of 400 high school students enrolled in a required computer literacy course featuring instruction in BASIC. Fewer than half of the boys and one-third of the girls reported that they liked working with computers or programming (Lockheed et al.,1983).

There also is some evidence that courses featuring programming languages affect boys and girls differentially. For example, Collis (1984) found that girls' attitudes about computers were associated negatively with exposure to BASIC programming courses, whereas boys' attitudes were associated positively with them. Similarly, Miura (1984) found that seventh-grade girls in programming classes reported liking computers less than their male classmates, and were less likely to take a subsequent elective computing course.

There is considerable evidence that girls are less likely than boys to elect voluntarily to take programming courses. For example, transcript data from High School and Beyond (HS&B) show that 41% of those enrolled in programming classes in high school in 1982 were girls (Rock et al., 1985). Self-report data from the 1983-84 NAEP show that 31% of eighth-grade students who wrote programs frequently were girls (Lockheed, 1985b). On the other hand, girls who enroll in elective programming courses often have positive attitudes about computers (Chen, 1984).

By comparison, courses featuring applications software seem to appeal to students more than programming courses do. Although there is no published research that directly addresses this issue, we do know that students enrolled in applications courses report positive attitudes about computers. For example, in a study of 100 seventh-grade students enrolled in a computer literacy course featuring word processing and database managers, nearly two-thirds (63%) of the students agreed with the statement "I enjoy working with computers," and more than half (56%) disagreed with the statement "Computers are boring" (Lockheed, Gulovsen & Morse, 1985). Voluntary enrollments in non-programming classes also are more balanced by sex. In the HS&B data, for example, 51% of students enrolled in data processing and applications courses and 47% of students enrolled in computer literacy courses were girls (Rock et al., 1985). In the NAEP data, for all types of computer use, no sex differences in course enrollments were found for students at any of the three grades assessed (Lockheed, 1985a).

Model Projects

At present, neither research nor development efforts on applications software use in schools are abundant. However, change is imminent. Both national and local curricula have been designed around the use of applications software. Several projects have demonstrated creative and exemplary uses of microcomputer technology in educational settings, and at least one research program has begun to explore systematically some of the many issues related to student use of applications software. We describe briefly several model projects.

Model Curricula

One example of a precollege computer curriculum based on applications software is provided by the Federal Republic of Germany's Institute for Science Education computer literacy curriculum. This curriculum was developed through an open, process-oriented procedure known as the curriculum conference, and involved educators, students, and interested laypersons. Features of the curriculum include the following: (a) introduction of computer literacy as a course for eighth-grade students, limited to 60 class periods; (b) separation of introductory and advanced levels of computer literacy, with the introductory level focused on the student as a computer user; (c) teaching the use of the "software tools such as data management systems (e.g., dBase II)" rather than programming languages; and (d) a decision that at the upper secondary level "software tools can replace previously used programming languages for the application of microcomputers in a number of subjects" (Bosler, undated). The overall computer literacy course is divided into three components: use of the application system, thematically related project units, and problem solving with algorithmic method (Bosler & Frey, 1984; Bosler, Hampe, Wanke, & van Weert, 1985).

Given the decentralization of American education, it is not possible to adopt or implement a national computer literacy curriculum, or any other national curriculum, whether or not it is based on applications software. Several local systems, however, have adopted applications-based computer literacy curricula. One example is the seventh-grade computer education curriculum from the Edward Devotion School of Brookline, Massachusetts. The major objective of this course was to have students experience the computer as a useful tool. During the year, students learned to use both a word processor and a database manager and worked with several different simulation programs; no programming was taught. Students wrote articles and reports on the computer and collected and analyzed data.

Other projects have focused on applications to promote more equitable uses of computers. One such project, funded by the Department of Education, is "Practical Solutions to Overcoming Inequities in Computer Use" at the American Institutes for Research (Schubert, 1984, 1985). The goal of the project is to identify and define factors that limit access to computers for females and minorities. The project also seeks to develop potential solutions and identify sources of inequity in schools.

A second project funded by the Department of Education is "The Neuter Computer" of the Women's Action Alliance of New York (Sanders & Stone, 1985). The project's goals are to develop optimal strategies for computer use at the middle school level and to increase computer opportunities for, and use by, females. The project is intended to improve computer education for all students by designing a variety of computer activities and specific implementation strategies. These strategies describe what can be done in the classroom, school, and community, and what parents can do to improve computer education for the broader student population. Some of the suggestions include establishing a sign-up system so that males are less likely to dominate the laboratory during free time, or encouraging group interaction that is likely to enhance interest among females.

Model Programs

Several programs have explored how computers can be used effectively for various students at different grade levels. The programs provide models for how to integrate applications software into the extant curricula and increase female and minority participation in computer-related activities.

Computer tutors. One project that focuses on the use of applications and encourages minority participation is the Computer Tutors program, directed by Joan Targ, at the Jordan Junior High School in Palo Alto, California. All students, when they are not in class, have access to two classrooms of microcomputers. Students learn how to use complex word processors for their writing assignments, graphics packages, database managers, and spreadsheets. They are encouraged to make full use of the computer laboratory for any type of assignment that could be accomplished easily with the help of a computer.

To use the computer laboratory, students must have participated in the tutor program. Students are assigned to a computer and receive instruction from a same-sex, same-ethnic group peer. The intent is for the tutor to serve as a peer role model, thereby increasing the likelihood that the novice student will react positively to the computer. Each peer tutor has received instruction on providing support to the novice, increasing interest, and conveying information about the computer. Empathy is fostered because all of the tutors were once novices. Thus, students learn about computers and they learn to impart acquired knowledge to other students while using applications software that is integrated into their daily coursework.

The ETS/IBM Project. A second project that focused on the use of applications was the ETS/IBM Project, a model for effective use of computers in secondary schools (Cline, Bennett, Kershaw, Schneiderman, Stecher, & Wilson, 1986). The goals of the project were to develop ways to enhance the use of applications software and to explore the use of computers as a tool to promote learning across content areas and in extracurricular activities. A major component of the project was a three-state network of teacher-training institutes that provided support to, and training for, secondary school teachers. The institutes helped the teachers develop their own materials and curricula for a variety of subjects and activities. The teachers then implemented the newly developed applications uses in 89 demonstrated schools. The teacher-training institutes continued to provide support for the teachers following the initial training period. The project demonstrated how applications could be used effectively for a variety of educational activities at the secondary school level.

Research

Research on the effects of using computer applications has only just begun. Several studies on the difficulties with and effects of using applications software have been completed or are in progress at the Educational Technology Center at Harvard University. Researchers have sought to understand what difficulties students and teachers encounter while trying to integrate microcomputers into the ongoing curriculum (Lockheed et al., 1985), and have begun to study the cognitive consequences of using applications software (Lockheed, Gulovsen & Morse, 1985). These studies have been primarily descriptive, focusing on the observed difficulties encountered by learners of applications software. Much more work is needed to fully understand the issues.

Conclusions

We have noted a changing trend in precollege computer curricula toward an emphasis on applications-based courses. We also have outlined a hypothetical model of the cognitive consequences of programming and applications-based learning activities and presented several model computer projects. We now consider issues related to the implementation of such curricula and propose a research agenda to examine that implementation.

Implementation Issues

National survey results have provided a good understanding of the current status of educational computing (e.g., Becker, 1985). However, these results fail to describe the determinants and consequences of implementing a computer curriculum in school settings. The process of implementation should be documented and address the question: What are the factors involved in actually getting a well-structured computer curriculum into the classroom? Issues that must be addressed concern the need for (a) adequate teacher training; (b) administrators and curriculum specialists who are informed about the potentials of computers; (c) adequate physical resources (i.e., appropriately configured hardware and software); and (d) software targeted to the appropriate educational level. Each of these factors affects the way computer curricula are implemented.

Research Agenda: Applications-Based Courses

It is critical for future research not only to examine the cognitive demands and consequences of computer learning environments, but also to consider the changing definition of computer literacy and the corresponding modifications in computer curricula. Future research should provide systematic cognitive analyses of computer-based learning activities and explore the instructional implications of their adoption into school curricula. Such research should help elucidate the effects of computer learning environments and establish an empirical base from which policy recommendations can be made.

The curricular shift from programming to applications necessitates research on the cognitive, afffective, and social effects of using such software for instructional purposes. Because applications programs deal with generic functions such as data manipulation and word processing, they can be applied across subject areas (e.g., English, social studies) and integrated into extant curricula.

More important, however, a well-structured applications-based curriculum may not only foster the acquisition of higher cognitive skills, but also stimulate positive affective responses for students who heretofore have been disinterested in programming. If applications can produce positive cognitive effects, and also increase affective responses to computer science in a broader range of students, then it seems reasonable to focus computer instruction on such software.

Cognitive analyses of the applications would serve as the initial phase of the research, and the precursor to the implementation of an applications-based curriculum. Research needs to establish the theoretical and empirical links between programming and applications courses and the potential outcomes that are likely to result from different instructional orientations. More specifically, the research must examine the cognitive demands and consequences of learning from applications, and compare those findings with corresponding studies of programming. The social and affective effects of the curricula must be studied. Research also must examine the effects of curricula in the different computer learning environments to evaluate and demonstrate the potential of the instructional media.

In conclusion, a redefinition of computer literacy has been suggested. The knowledge of how to program does not necessarily assume that individuals are computer literate. Moreover, the intent of precollege instruction is not to create a group of expert programmers, but rather to provide basic knowledge about how to use computers in everyday life. An applications-based curriculum is likely to focus on skills that are applicable in a variety of real-life situations, and thereby serve as an appropriate introduction to computers for a broad range of students.

References

Anderson, R. E., Welch W. W., & Harris, L. J. (1984). Inequities in opportunities for computer literacy. *The Computing Teacher, 11* (8), 10–12.

Atwood, M. E., & Ramsey, H. R. (1978). *Cognitive structures in the comprehension and memory of computer debugging* (Tech. Rep. No. TR–78–A210). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Becker, H. J. (1983a). *School uses of microcomputers* (Issue 1). Baltimore, MD: The Johns Hopkins University, Center for Social Organization of Schools.

Becker, H. J. (1983b). *School uses of microcomputers* (Issue 2). Baltimore, MD: The Johns Hopkins University, Center for Social Organization of Schools.

Becker, H. J. (1983c). *School uses of microcomputers* (Issue 3). Baltimore, MD: The Johns Hopkins University, Center for Social Organization of Schools.

Becker, H. J. (1984). *School uses of microcomputers* (Issue 5). Baltimore, MD: The Johns Hopkins University, Center For Social Organization of Schools.

Becker, H. J. (1985, May). *School uses of computers: Results from the second survey.* Paper presented at Educational Testing Service Sociology of Education Seminar, Princeton.

Bosler, U. (undated). *The curriculum conference—A procedure for the formulation of a computer literacy curriculum in the 8th grade.* Kiel, Federal Republic of Germany: IPN–Institute for Science Education.

Bosler, U., & Frey, K. (1984). *Curriculumkonferenz zum thema grundbildung informatik-programmheft.* Kiel, Federal Republic of Germany: IPN, Polyskript.

Bosler, U., Hampe, W., Wanke, I., & van Weert, T. J. (1985). *Grundbildung informatik-Anregungen fur ein curriculum.* Kiel, Federal Republic of Germany: IPN, Polyskript. Geplant: Fruhjahr 1985 auch Stuttgart: Metzler.

Boyer, E. L. (1983). *High school: A report on secondary education in America.* New York: Harper and Row.

Chen, M. (1984). *Gender differences in computer uses and attitudes: A survey of five Bay Area high schools.* Stanford, CA: Stanford University, Institute for Communication Research.

Clements, F. J. (1981, April). Affective considerations in computer-based education. *Educational Technology, 21,* 28–32.

Cline, H. F., Bennett, R. E., Kershaw, R. C., Schneiderman, M. B., Stecher, B., & Wilson, S. (1986). *The electronic schoolhouse.* Hillsdale, NJ: Erlbaum.

College Board. (1974). *National college-bound seniors, 1973–74.* New York: The College Entrance Examination Board.

College Board. (1983a). *Academic preparation for college.* New York: The College Entrance Examination Board.

College Board. (1983b). *National college-bound seniors, 1983.* New York: The College Entrance Examination Board.

College Board. (1984). *National college-bound seniors, 1984.* New York: The College Entrance Examination Board.

College Board. (1985). *National college-bound seniors, 1985.* New York: The College Entrance Examination Board.

Collis, B. A. (1984). *The development of an instrument to measure attitudes of secondary school males and females toward computers.* Unpublished doctoral dissertation, University of Victoria, Victoria, British Columbia.

Corno, L., & Mandinach, E. B. (1983). The role of cognitive engagement in classroom learning and motivation. *Educational Psychologist, 18,* 88–108.

Demarest, M., & McKenzie, J. (1984). *Probing the computer gap.* Unpublished manuscript.

Fetler, M. (1985). Sex differences on the California statewide assessment of computer literacy. *Sex Roles, 13,* 181–192.

Glaser, R. (1984). Education and thinking. *American Psychologist, 39,* 93–104.

Goor, J. (1982). *Student use of computers in schools.* Washington, DC: National Center for Education Statistics.

Hunter, B. (1983). *My students use computers: Computer literacy in the K–8 curriculum.* Reston, VA: Reston.

Kurland, D. M., Mawby, R., & Cahir, N. (1984, April). *The development of programming expertise in adults and children.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans.

Lawton, J., & Gerschner, V. T. (1982). A review of the literature on attitudes towards computers and computerized instruction. *Journal of Research and Development in Education, 16*(1), 50–55.

Linn, M. C. (1985). The cognitive consequences of programming instruction in classrooms. *Educational Researcher, 14*(5), 14–16, 25–29.

Lockheed, M. E. (1985a). *Determinants of student computer use: An analysis of data from the 1984 National Assessment of Educational Progress.* Princeton, NJ: Educational Testing Service.

Lockheed, M. E., (1985b). Women, girls and computers: A first look at the evidence. *Sex Roles, 13,* 115–122.

Lockheed, M. E., Gulovsen, J., & Morse, S. (1985). *Observations of computer-using students.* Cambridge, MA: Harvard Graduate School of Education, Educational Technology Center.

Lockheed, M. E., Gulovsen, J. P., & Morrison, D. (1985). *Student use of applications software* (Tech. Rep. 85–3). Cambridge, MA: Harvard Graduate School of Education, Educational Technology Center.

Lockheed, M. E., Nielson, A., & Stone, M. K. (1983). Sex differences in microcomputer literacy. In D. Bonnette (Ed.), *Proceedings of NECC/5, National Educational Computing Conference 1983.* Silver Spring, MD: IEEE Computer Society Press.

Mandinach, E. B. (1985, October). *Cognitive consequences of microcomputer applications.* Paper presented at the Computers and Complex Thinking Conference, Washington, DC.

Miura, I. T. (1984). *Processes contributing to individual differences in computer literacy.* Unpublished doctoral dissertation, Stanford University.

National Assessment of Educational Progress 1983–84 Public-Use Data Tapes Version 3.0 (1985). Prepared for the National Institute of Education by the National Assessment of Educational Progress. Princeton, NJ: Educational Testing Service.

National Science Board Commission on Precollege Education in Mathematics, Science and Technology. (1983). *Educating Americans for the 21st Century.* Washington, DC: National Science Foundation.

Patterson, J. H., & Smith, M. S. (1985, October). *The role of computers in higher order thinking.* Paper presented at the Computers and Complex Thinking Conference, Washington, DC.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology, 2,* 137–168.

Rock, D., Ekstrom, R., Goertz, M., & Pollack, J. (1985). *A study of excellence in high school education: Longitudinal study, 1980–82.* Unpublished report under review by the National Center for Education Research.

Sanders, J. S., & Stone, A. (1985). *The neuter computer: Why and how to encourage computer equity for girls.* New York: Women's Action Alliance, Inc.

Schubert, J. G. (1984, April). *Females and microcomputer use in school: Some insights into traditional patterns.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans.

Schubert, J. G. (1985, April). *Cooperative learning, equity, and computer learning.* Paper presented at the annual meeting of the American Educational Research Association, Chicago.

Tucker, M. S. (1983). *Computers in the schools: The federal role.* Unpublished manuscript.

Wilder, G., Mackie, D., & Cooper, J. (1985). Gender and computers: Two surveys of computer-related attitudes. *Sex Roles, 13,* 215–228.

Williams, F., & Williams, V. (1984). *Microcomputers in elementary education: Perspectives on implementation.* Belmont, CA: Wadsworth.

Wright, D. A., Melmed, A., & Farris, E. (1984). *Instructional use of computers in public schools.* Washington, DC: National Center for Education Statistics.