

LABORATORY FOR
COMPUTER SCIENCE



MASSACHUSETTS
INSTITUTE OF
TECHNOLOGY

AD-A209126
MIT/LCS/TR-433

COMPUTER LITERACY EDUCATION

Ronni Lynne Rosenberg



January 1989

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

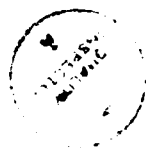
REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MIT/LCS/TR-433			5. MONITORING ORGANIZATION REPORT NUMBER(S) N00014-83-K-0125	
6a. NAME OF PERFORMING ORGANIZATION MIT Laboratory for Computer Science		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION Office of Naval Research/Department of Navy
6c. ADDRESS (City, State, and ZIP Code) 545 Technology Square Cambridge, MA 02139			7b. ADDRESS (City, State, and ZIP Code) Information Systems Program Arlington, VA 22217	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION DARPA/DOD		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Blvd. Arlington, VA 22217			10. SOURCE OF FUNDING NUMBERS	
			PROGRAM ELEMENT NO.	PROJECT NO.
			TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) Computer Literacy Education				
12. PERSONAL AUTHOR(S) Rosenberg, Ronni Lynne				
13a. TYPE OF REPORT Technical		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) 1989 January
15. PAGE COUNT 160				
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Computer literacy, education, Science Technology and Society	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This thesis presents a study of computer literacy in schools, informed by the perspectives of education professionals and computer professionals. The study examines the traditional arguments of computer enthusiasts for computer education and analyzes their underlying justifications, giving counterarguments where appropriate. Examples are taken from classroom observation, interviews, and the literature. Many well intentioned schools and dedicated teachers are striving to overcome the serious flaws that characterize computer-literacy education: vague goals, inadequate hardware, bad software, and poor training. But while computers can be useful tools, mandatory computer education is unjustified. Computer literacy as a fundamental skill (like reading and writing) is oversold, misapplied, basically trivial in many applications, and not demonstrably up to its claims for education. In many cases, enthusiasm for computers and education has less to do with the educational value of computers than with a complex web of social pressures on educators. Moreover, the euphoria about computers in schools can be a smokescreen, —> (continued)				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Judy Little, Publications Coordinator			22b. TELEPHONE (Include Area Code) (617) 253-5894	22c. OFFICE SYMBOL

19.

diverting attention from fundamental educational problems, such as the demoralizaing environments in which many primary- and secondary-school teachers work, or the lack of real literacy among many graduates. Such problems do not have technological solutions. Thesis (700/111)

Accession For		
NTIS GRA&I	<input checked="" type="checkbox"/>	
DTIC TAB	<input type="checkbox"/>	
Unannounced	<input type="checkbox"/>	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		



Computer Literacy Education

Ronni Lynne Rosenberg

MIT / LCS / TR-433
January 1989

© Massachusetts Institute of Technology 1989

This report describes research done at the Laboratory of Computer Science of the Massachusetts Institute of Technology. This work was sponsored in part by the Defense Advanced Research Projects Agency under contract number N00014-83-K-0125. The author was supported in part by a GTE Graduate Fellowship and an Alfred Kiel Fellowship for the Wiser Uses of Science and Technology. This report was originally published as the author's doctoral dissertation.

Computer Literacy Education

Ronni Lynne Rosenberg

Technical Report MIT / LCS / TR-433
January 1989

*MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139*

Abstract

This thesis presents a study of computer literacy in schools, informed by the perspectives of education professionals and computer professionals. The study examines the traditional arguments of computer enthusiasts for computer education and analyzes their underlying justifications, giving counterarguments where appropriate. Examples are taken from classroom observation, interviews, and the literature. Many well intentioned schools and dedicated teachers are striving to overcome the serious flaws that characterize computer-literacy education: vague goals, inadequate hardware, bad software, and poor training. But while computers can be useful tools, mandatory computer education is unjustified. Computer literacy as a fundamental skill (like reading and writing) is oversold, misapplied, basically trivial in many applications, and not demonstrably up to its claims for education. In many cases, enthusiasm for computers and education has less to do with the educational value of computers than with a complex web of social pressures on educators. Moreover, the euphoria about computers in schools can be a smokescreen, diverting attention from fundamental educational problems, such as the demoralizing environments in which many primary- and secondary-school teachers work, or the lack of *real* literacy among many graduates. Such problems do not have technological solutions.

Acknowledgements

My first debt is to the people who generously shared their time with me to discuss their work in and thoughts about computer-literacy education. I am most grateful to the educators – primary- and secondary-school teachers, administrators, computer-laboratory directors, and computer coordinators – who made time in their hectic schedules to educate me about their profession.

Many computer professionals also contributed to this work. I especially thank the RISKS community: those who are part of the electronic digest “Forum on Risks to the Public in Computers and Related Systems,” moderated by Peter G. Neumann.¹ Computer Professionals for Social Responsibility (CPSR) was an ongoing source of encouragement for me and, more generally, for work in the social implications of computer technology.²

Material support for my doctoral program came from several sources at MIT: the Laboratory for Computer Science, the Provost's Office, and the Office of the Dean of Engineering. Professor Michael Dertouzos, Director of the Laboratory for Computer Science, was instrumental in pulling together financial support for me. My work was sponsored in part by the Defense Advanced Research Projects Agency under contract number N00014-83-K-0125, a GTE Graduate Fellowship, and an Alfred Kiel Fellowship for the Wiser Uses of Science and Technology.

I thank MIT's Department of Electrical Engineering and Computer Science and the committee that supervised my interdisciplinary program of study for giving me the opportunity to work outside traditional disciplinary boundaries. The committee consisted of Kenneth Keniston, Director of MIT's Program in Science, Technology, and Society, Ned Block, Professor of Philosophy, and Joseph Weizenbaum, Professor Emeritus of Computer Science, who chaired the committee. I owe a special debt to Joe Weizenbaum for unfailingly focusing my attention on the heart of the matter.

¹See Appendix B for more information on RISKS.

²CPSR, P.O. Box 717 Palo Alto, CA 94301, 415-322-3778

Contents

1	Introduction	7
1.1	Research Plan	13
1.2	Guide to this Report	15
2	How is Computer Literacy Taught?	17
2.1	Terminology and Jargon	20
2.2	Hardware	23
2.3	Software	25
2.4	Programming	31
2.5	Jobs in Computing	38
2.6	Social Impacts of Computers	38
2.7	Powers and Limits of Computers	41
3	Educators' Perspectives	45
3.1	Hardware and Software	47
3.2	Funding	55
3.3	Training: Little Change from Kindergarten through Graduate School	57
3.4	Priorities	63
3.5	Pressures to Use Computers	68
3.6	Technology-Driven Education: 'A Bandaid on What's Wrong with the School System'	74
4	Computer Professionals' Perspectives	77
5	Why Is Computer Literacy Taught?	89
5.1	The Argument of Jobs	90
5.2	The Argument of Mental Discipline: 'Wheaties of the Mind'	98
5.3	The Argument of Informed Citizens	109
5.4	Other Reasons	113
6	Conclusions	118
6.1	Future Work	128
7	Bibliography	130

A Survey on Computer Literacy (for Teachers)	148
B Survey On Computer Literacy (for Computer Professionals)	150
C Curriculum Vitae	157

Chapter 1

Introduction

Computing is the predominant technology of this age. Computers are at the heart of modern organizations – banks, factories, communications companies, transportation systems, and the military – and transactions with computers are part of daily life. Decisions about how computers should – and should not – be used will be made increasingly by people who are graduates of an educational system that incorporates “computer literacy” as a fundamental component. The wisdom of these decisions will depend largely on the *quality of that education.*

In this paper, I present a study of the phenomenon of computer literacy in schools, informed by the perspectives of both education professionals and computer professionals. While computer literacy is poorly defined, it is the most common term used to refer to a body of information about computers which, according to its enthusiasts, will transform today’s students into the productive workers, informed citizens, and wise decision-makers of our “information society.” Computer literacy, defined in a variety of ways, is becoming a standard part of the curriculum of primary and secondary schools throughout the United States.

The use of the term “literacy” in conjunction with computers is not accidental; it reflects the view of some advocates that computer-related skills are as important as is the ability to read and write. Use of the term “computer literacy” implies the existence of a group of “computer illiterates,” who by definition are at an unspecified but nonetheless serious disadvantage compared to computer-literate people. “The device,” insists Charles Suhor, deputy executive director of the National Council of Teachers of English, “is raw

propaganda, and does no service to reading and writing (which must now, it seems, be redundantly called 'print literacy')."¹ Previous technologies (e.g., radio, television, and telephone) also generated much enthusiasm in the educational community, but few people suggested that it was necessary for everyone to learn about them. Unlike computers, these earlier technologies were not compared in influence to print, and they were not elevated to the status of "literacy."

In contrast, educational computing enthusiasts believe that there is something so special about computer literacy, that it should be added to educators' "short lists" of basic skills, along with, for instance, English, mathematics, science, and social studies. The much-publicized report "A Nation at Risk" lists computer science as one of "Five New Basics" that all high-school students should be required to study.

In response, schools have bought more computers, particularly during the past few years. By many measures, computing in schools has spread rapidly. Most surveys of computers in schools reveal steady increases over the past few years in, for instance, the computer/student ratio in many schools. The most detailed study to date of microcomputer use in schools found that as of spring 1985, 86% of U.S. schools had at least one computer for instruction.² This figure was up from 53% reported for January 1983.³ Documenting the degree to which computers are fast becoming a classroom fixture, this same study found that from 1983 to 1985:

- The number of computers in schools jumped from 250,000 to 1 million.
- Three-fourths of the 33,000 schools that had not previously used computers began to do so.
- The proportion of elementary schools with 5 or more computers jumped from 7% to 54%, and the proportion of secondary schools with 15 or more computers increased from 10% to 56%.

¹Tom Snyder and Jane Palmer, *In Search of the Most Amazing Thing: Children, Education, and Computers* (Reading, Massachusetts: Addison-Wesley Publishing Company, 1986) 52-53.

²Henry Jay Becker, project director, "Instructional Uses of School Computers" (Baltimore, Maryland: Center for Social Organization of Schools, The Johns Hopkins University, 1986-1987) series of six newsletters.

³Snyder and Palmer, 31.

- The time that computer-using high-school students spent at terminals each week nearly doubled, to 90 minutes.
- The number of students using computers jumped from 4.5 million to 15 million.
- By 1985, one million pre-college students in the United States took computer programming courses.⁴

Along with an increase in the presence of computers in schools, there has been an increase in the use of those computers to teach computer literacy. The National Assessment of Educational Progress (NAEP) does regular, large-scale studies of primary and secondary education throughout the country, for the Federal government. In 1980, they found 15% of elementary schools and 50% of secondary schools offering instruction in the use of computers. By 1985, these figures climbed to 82% of elementary schools and 93% of secondary schools.⁵ According to the 1985 study, school "computers are used almost exclusively to teach about computers";⁶ that is, to teach computer literacy.

When *Electronic Learning* surveyed state education agencies in 1983, they found 17 states with laws requiring or recommending computer-literacy instruction in K-12 (kindergarten through twelfth grade) schools. Similar bills were under consideration in a good many more states.⁷ In another survey (conducted in November 1985), 96% of the respondents – primarily classroom teachers, computer coordinators, and administrators – said that their schools offered instruction in computer literacy (mostly stand-alone courses, taught separately from the rest of the curriculum).⁸

While the non-military portion of the federal budget has been slashed, and the educational system is in relatively poor financial shape, money has somehow been found for computers. Computer classes are added to schools as fast as money to buy computers can

⁴Becker, "Instructional Uses of School Computers."

⁵National Assessment of Educational Progress, "Computer Competence: The First National Assessment" (Princeton, New Jersey: Educational Testing Service, 1988) 7.

⁶Ibid., 3.

⁷T. Spain, "How State Governments are Helping to Define Computer Literacy," Article 10b in Terence R. Cannings and Stephen W. Brown, eds., **The Information Age Classroom: Using the Computer as a Tool** (Irvine, California: Franklin, Beedle & Associates, 1986) 53.

⁸"Reader's Survey Results: What Is Computer Literacy?" *Classroom Computer Learning* (March 1986) 53.

be squeezed from strained school budgets (in some cases, even faster, as some schools decide to teach computer classes although they cannot afford to buy any computer.). The incorporation of computers represents a frenzy of change in the normally slow, staid educational system.

Continued large investments will be required for schools to address the many problems that educators cite, when asked about their current use of computers:

- Nationwide, the average school still has only one computer for every 40 students (as of the last large-scale survey). And the computers that schools can afford are at the inexpensive, weak end of the personal-computer spectrum.
- Almost everyone agrees that the vast majority of so-called educational software is awful and does not meet teachers' curriculum needs.
- Teacher training in the use of computers is abysmal, when it exists at all.
- After spending all their money on equipment, some schools do not have adequate funds leftover for repairs or other support services.

Moreover, large expenditures on computers reflect difficult tradeoffs for schools. More money for computers means less money for teachers and teacher aides; less money for books, maps, and laboratory supplies; less money for facility maintenance; and so on. More time for computer classes, curriculum development, class preparation, teacher training, and so on means less time for other subjects. Are these tradeoffs justified?

Critical evaluation – evaluation of how and why computer literacy is taught – is in short supply. The increasingly focused attention on computers in education has both resulted from and contributed to a veritable frenzy of equipment purchases and curriculum changes in schools. Although computer-literacy advocates agree that there are important, unsolved problems, the momentum to increase the use of computers in schools remains unabated, and the education literature remains wildly optimistic about computers. The equipment purchases and curriculum changes are taking place in an atmosphere almost completely devoid of critical evaluation. Moreover, there has been little attempt

to inform these educational changes with the insights of people who are already computer literate: computer professionals. (To the extent that "computer literacy" means anything, it must apply to computer professionals.)

The enthusiasm about computers that infects the education literature is almost uniformly uncritical. Most papers simply describe the format of a class or piece of software, with little evaluation beyond the frequent complaints that there are too few machines and the software is inadequate. Despite the lack of evaluation, the assumption that computer education is worthwhile, even essential, is rarely challenged. Most articles and individuals assert benefits of computer literacy while giving no analysis or justification whatsoever.

The most frequently cited reasons for teaching computer literacy are these:

1. Computer job skills are needed by everyone.
2. Computing is "good for the mind." It improves general thinking skills.
3. Computer literacy is a fundamental life skill. Every informed citizen in today's society "must" know about computers.

In Chapter 5 ("Why Is Computer Literacy Taught?"), I examine in detail the basis for these reasons.

Another reason for teaching about computers, often cited by teachers, is pressure from parents, administrators, and the media. Because of the widely touted benefits of computer literacy, it is sometimes easier for schools to obtain money for computers (e.g., state grant funds or money raised by parents) than for anything else (such as more teachers or books). A school's initial hardware may be purchased at a substantial "educational discount" or may even be donated. Schools are fertile ground for marketing departments of computer companies. By providing today's students with a particular brand of hardware and software, computer companies are establishing ties to tomorrow's computer buyers.

The reasons offered for teaching computer literacy do not stand up to close scrutiny. When schools decide to offer computer-literacy courses, they do so despite the almost total absence of well grounded reasons. While promoters of computer-literacy education

have been quick to make promises about its educational benefits, they are promises that have not yet been fulfilled in practice, nor adequately addressed through research. Instead, the field is being driven by largely abstract debates on *potential* educational value. Benefits are expressed in the future tense, and there is always a technical explanation given for why these benefits have not yet been achieved; e.g., not enough machines.

A common theme in arguments for computer-literacy education is the alleged necessity to know about computers, to function in today's society:

"As Senator Lautenberg said upon the introduction of the Computer Education Assistance Act of 1984, 'Our competitive position in the international economy is dependent on the ability of this country to produce well-educated, skilled and creative workers.' Computer education programs can significantly improve our ability to do so by providing powerful tools to improve the quality of education in all subjects and by introducing children to the technology of the future. ... computer education cannot be thought of as an optional course or a luxury, it is nothing less than a necessity."⁹

In one local school district, the computer coordinator argues that "The well being of today's students could depend heavily upon how competent they are in understanding and using computer technology."¹⁰ A computer-literacy textbook says this: "You do not have to depend on others You do not have to trust experts. You are *literate*. ... You are in control. When you become computer literate, you will know two important things: (1) What things a computer *can* do and (2) how to tell a computer to do the things *you* want it to do."¹¹

This is typical of the strong rhetoric of those arguing in favor of widespread computer education. But is this rhetoric justified? Does what we know about teaching and learning

⁹Paul Bonner, "Computers in Education: Promise and Reality," Article 5 in Terence R. Canning and Stephen W. Brown, eds., **The Information Age Classroom: Using the Computer as a Tool** (Irvine, California: Franklin, Beedle & Associates, 1986) 25.

¹⁰Joanne Krepelka, "Action Plan for Computer Technology, 1986-87" (Cambridge, Massachusetts: Cambridge Public School System, 1986), computer coordinator's report, ii.

¹¹Arthur Luehrmann, **Computer Literacy: A Hands-On Approach** (U.S.A.: McGraw-Hill, Inc., 1983) 1.

computer literacy support what is said about the value of computer-literacy education? The key question is: What is behind all the attention to computer literacy in schools?

1.1 Research Plan

Against the widespread background of optimism for the computerization of education, the goal of my research is to assess computer-literacy education in a study that differs from previous work in two ways. First, it is informed by the perspectives of not only education professionals, but also computer professionals, the most computer-literate group in society. Second, it takes a critical look at computer literacy, asking for justification for claims about the benefits of computers in schools.

I collected data from several sources:

- **Literature review.** I read extensively in the education literature and, where appropriate, the computing literature. A current bibliography of about 250 computer-literacy items is in Chapter 7. It ranges from academic to popular works and includes classroom materials. The literature review provided much of the information about the status of computer-literacy education – the facts about curriculum, hardware, software, and so on.

My literature review included about three dozen classroom materials, ranging from computer-literacy textbooks to unpublished school, district, and state computer-literacy curriculum guides. In addition, many individual classes are described in articles in the education literature. I found a great deal of similarity among all the computer-literacy class descriptions I reviewed.

- **Classroom observation.** I observed two computer-literacy classes (both in Massachusetts): one middle-school class for eighth-grade students and one graduate class for teachers, administrators, and computer coordinators. I observed other educational uses of computers in additional schools. Classroom observation provided more information about the status of computer-literacy classes.

Also, what I observed yielded important insights that were not revealed in the

literature. For example, my classroom observations made clear the great extent to which computer-literacy education is vocational education, preparation for menial clerical jobs. Much of it concentrates on the mechanics of how to operate machines, handle floppy disks, and run printers.

What I observed in schools highlights the problem of inadequate teacher training: In the graduate class for teachers and administrators, a computer operating system was defined incorrectly as a program that "directs the flow of electricity." From this class, two dozen teachers went forth with some fundamental misconceptions about how computers work.

And what I observed reveals a danger of trying to make people "computer literate" by exposing them to "toy" computers: In one class, students were assured wrongly that the people who create complex computer programs understand and control those programs. Initially, the students worried about potential negative impacts of computers. Based on this class discussion, they reassured themselves that there are computer experts somewhere who control the large, complicated systems on which everyone depends. This conclusion is comforting but dangerously misleading.

- **Interviews.** I conducted eleven interviews with computer coordinators, computer-literacy teachers (including the teachers of the two classes mentioned above), and other teachers (primary and secondary) who were acquiring computer-literacy training in preparation for using computers in their classrooms in a variety of ways. Interviews (and surveys, below) provided additional status information about teacher training and school support for computer-literacy classes. In these formal and detailed interviews, I probed
 - teachers' beliefs about the value of teaching computer literacy and the foundations of these beliefs,
 - teachers' perceptions of what schools should be teaching and how in their view these are or are not supported by computers, and
 - how teachers estimate the hidden costs of fitting computer time into already-full classes and school days.

Many informal discussions with other teachers also probed for evidence to support the reasons given for teaching computer literacy.

- **Surveys.** I surveyed computer-literacy teachers and administrators who were themselves the students of the graduate class that I observed. In the survey, I probed many of the same issues as in the interviews, but over a wider audience (and in less depth). Since the teachers I surveyed work in many different schools and school districts, I was able to use the survey to obtain information about the status of computer-literacy education in many more schools than I could observe.

I distributed a survey to computer professionals over an international computer network. Access to the network enabled me to obtain responses from many people. This was an important source of information about the views of computer technologists on computer-literacy education. I solicited people's views on how computer literacy *should be* taught (if at all) and why, as well as their reactions to information about how it *is* taught.

Questions for each survey are listed in appendices.

Data analysis followed data collection. By "data analysis," I mean interpretive analysis – analysis that is at once qualitative and disciplined. The approach to information gathering was largely an ethnographic one, starting with the premise that what is to count as "fact" about computer literacy depends partly on the observer's viewpoint.¹² My objective has been to present two relevant viewpoints in one study.

1.2 Guide to this Report

The remainder of this thesis is organized as follows:

¹²See John Van Maanen, **Tales of the Field: On Writing Ethnography** (Chicago, IL: The University of Chicago Press, 1988). Eric Bredo and Walter Feinberg, eds., **Knowledge and Values in Social and Educational Research** (Philadelphia, PA: Temple University Press, 1982). Paul Rabinow and William M. Sullivan, eds., **Interpretive Social Science: A Reader** (California: University of California Press, 1979).

- Chapter 2. How Is Computer Literacy Taught?

This chapter describes in detail the topics commonly taught in computer-literacy classes. The variety of definitions of computer literacy are presented, including definitions implicit in computer-literacy course descriptions.

- Chapter 3. Educators' Perspectives

This chapter discusses computer-literacy education from the viewpoint of primary- and secondary-school educators. I use information from interviews, classroom observation, my written survey, and the education literature. This chapter also examines the environment surrounding computer-literacy education in primary and secondary schools: school requirements, teacher training, hardware (availability, quality, equity in access, maintenance), software (overall quality, types of programs), time (how is computer literacy fit into the school day), and money (how do schools afford computers, computer teachers, and computer coordinators).

- Chapter 4. Computer Professionals' Perspectives

This chapter discusses computer-literacy education from the viewpoint of computer professionals, using information from discussions and my electronic survey. Taken together, Chapters 3 and 4 document a wide divergence of views on computer literacy held by educators and by computer professionals.

- Chapter 5. Why Is Computer Literacy Taught?

Why do computer-literacy advocates believe that computer literacy should be taught? This chapter presents the traditional arguments for computer-literacy education and examines the basis for these arguments.

- Chapter 6. Conclusions

- Chapter 7. Bibliography

- Appendix A. Survey On Computer Literacy (for Teachers)

- Appendix B. Survey On Computer Literacy (for Computer Professionals)

- Appendix C. Curriculum Vitae

Chapter 2

How is Computer Literacy Taught?

Although computer literacy is fast becoming a standard part of the curriculum of primary and secondary schools throughout the United States, there is little agreement on a definition of computer literacy or the goal of computer-literacy classes. Many definitions are circular: "computer literacy is education about computers" or "computer literacy is whatever one needs to know about computers":

"The term computer literacy can be considered to mean the minimum knowledge, know-how, familiarity, capabilities, abilities, and so forth, about computers essential for a person to function well in the contemporary world."¹

"The MECC [Computer Literacy Study] authors responded that computer literacy is whatever knowledge and skills the average citizen needs to know about computers"²

"[T]he National Center for Educational Statistics (1983) assembled a panel of experts to define this elusive concept and guide subsequent national survey work. The ten-member panel settled on the following: 'Computer literacy may defined as whatever a person needs to know and do with computers in order to function competently in our information-based society.'³

¹ Alfred Bork, *Personal Computers for Education*, New York: Harper & Row, 1985, 33.

² "Mission: Define Computer Literacy. The Illinois-Wisconsin ISACS Computer Coordinators' Committee on Computer Literacy Report (May 1985)," *The Computing Teacher* (November 1985) 10.

³ Roy M. Gabriel, "Assessing Computer Literacy: A Validated Instrument and Empirical Results," *AEDS Journal*, 18, 3 (Spring 1985) 154.

Typical definitions contain vaguely important-sounding goals; e.g., "knowing how to use a computer" or "running a program." The definition of computer science put forth by the National Commission on Excellence in Education includes the following goals: "understand the computer as an information, computation, and communication device; ...; and understand the world of computers, electronics, and related technologies."⁴ In the classroom, vague goals usually translate into reducing the subject matter to a set of mechanical skills and superficial treatment of relevant as well as irrelevant issues.

Many course descriptions reflect instruction that is little more than a presentation of lists for students to memorize. A New York City guide to computer literacy lists "performance objectives" for a module on "Computers In Our Lives"; typical objectives are these: "State at least one use of the computer in the home. State at least one use of the computer in the school. State at least one use of the computer in our society. Describe at least three different computer careers."⁵ A study of computer-literacy course objectives was done by the Minnesota Educational Computing Consortium (MECC), a prominent distributor of classroom software. Of 63 objectives listed in the study,

"Fifty-one objectives involve nothing more than student acceptance of *hearsay* about computing, such as might be acquired from a book or from being told by the teacher. ... Altogether, there are eleven objectives that start with 'Identify,' twenty-one that start with 'Recognize,' and three that start with 'Determine.' In every such case little real change occurs if these verbs are replaced by the single word *Remember*. Indeed, the corresponding test items on the MECC assessment instrument require only that the student remember the 'right' set of words that goes with that objective."⁶

The core of most computer-literacy classes is threefold: computer terminology, hardware, and software. Many courses also incorporate programming, jobs in computing, or

⁴National Commission on Excellence in Education, "A Nation at Risk: The Imperative for Education Reform" (Washington, D.C.: U.S. Government Printing Office, April 1983) 26.

⁵Computer Literacy: Elementary Grades (Brooklyn, New York: Board of Education, City of New York, 1983) 147.

⁶Arthur Luehrmann, "Computer Literacy - What Should It Be?" *Mathematics Teacher*, 74, 9 (December 1981) 684.

social impacts of computers (including ethical and legal issues). Each topic is examined in more detail below.

Most arguments for computer-literacy education are based on a belief that computing is such a special technology, learning about it is as important as learning to read and write. However, the history of this allegedly momentous technology is usually not considered sufficiently important to be included in the curriculum. Where history is included, it may consist of little more than a "photo essay," a series of photographs of different types of computers. Most history components of computer-literacy classes do not consider the historical *context* in which computers were and are developed; e.g., the close relationship between computer technology and the requirements of the military. One computer-literacy curriculum guide states the following: "Students will be able to describe why the computer might have been invented (to help man extend the use of his brain and keep track of large numbers of things)."⁷ In fact, computers were invented (simultaneously in several different countries) to calculate bomb trajectories, break enemy codes, assist in urgently needed military-aircraft engineering calculations, etc.

In a number of definitions of computer literacy, key computing concepts are not distinguished from the mechanics of computer operation. One elementary-school computer-literacy curriculum lumps 13 topics, including programming, keyboard use, and "integration of computers into daily life," at a single level of hierarchy.⁸ Although many computer-literacy enthusiasts stress that students should be taught general concepts, most computer-literacy courses emphasize particular machines, programs, and languages. Where computer-literacy teachers are trained at all, their training also focuses on operating specific hardware and software. Given the great rate of change in the field of computers - where particular equipment and ways of doing things become

⁷Center for Learning Technologies, State Education Department, *Computer Literacy: An Introduction* (Albany, New York: The University of the State of New York, 1983) 77. Quoting Pat Walkington, "Curriculum Guide for Beginning Computer Literacy and Programming in BASIC," Phoenix, Arizona.

⁸Robert Calfee, "Computer Literacy and Book Literacy: Parallels and Contrasts," *Educational Researcher* (May 1985) 12.

obsolete quickly – machine-specific and program-specific education about computers is inappropriate.

Although computer literacy is perceived by some advocates as being as important as “print literacy,” computer-literacy education only skims the surface of knowledge about real-world computing. I did not encounter any computer-literacy courses or classroom materials that accurately present the computer abstractly or concretely as a symbol-processing device, discuss computational powers and limits, or consider in any depth the benefits and risks associated with important computer applications. However, I did encounter courses that explicitly “teach” students to “have positive attitudes about computers.” Such instruction discourages critical individual examination of the impacts of computers.

In contrast, computer scientists’ definitions of computer literacy are considerably more detailed than any computer-literacy course description. They scorn the emphasis on mechanical skills that characterizes computer-literacy classes. Computer professionals’ views on computer literacy (discussed in Chapter 4) are completely at odds with how computer literacy is taught.

2.1 Terminology and Jargon

Much of the time in computer-literacy classes is spent on jargon. In 1983, at Pepperdine University’s Computer Literacy Institute, educators from throughout the country decided that computer literacy must include “having a working knowledge of computer terminology.”⁹ Public-school teachers in the state of Virginia are required to define selected computer terms, in order to be certified in basic computer literacy.¹⁰

Fluency in a little jargon enables people to read magazine articles about computer technology with some comprehension, and this falsely nurtures their feeling of belonging in the “information society.” My colleagues in the computer profession laugh and nod

⁹James Penrod and Nancy Hughes, “Computer Literacy: Synopsis Of An Institute,” *Cause/Effect* (November 1983) 3.

¹⁰Chris Pipho, “The Computer Literacy Dilemma in the Public Schools,” *Educational Horizons* (Spring 1985) 101.

knowingly when I mention people who, having read the latest *Time* magazine article about computers, thrust themselves excitedly into a conversation about computers, confident that they can "talk knowledgeably" about them. In the classroom, exposure to terminology is often taken as sufficient to satisfy the common course goal of "having an awareness about computers."

How the computer is defined to people who are not yet computer literate is very important. Consider the following definitions from computer-literacy books:

- "A computer is a problem-solving machine."¹¹
- "A computer is an electronic machine that solves problems or answers questions."¹²
- "A computer ... is a machine that can handle large amounts of information and work with amazing speed."¹³
- "A computer is an electronic tool that helps people do many different things faster, easier, or better."¹⁴
- "A computer is a person or thing that computes; it is also an electronic machine that is used to perform specific kinds of knowledge work. It can accept information, process that information by carrying out mathematical operations at high rates of speed, and produce a logical result. ... Knowledge Work: Knowledge is essentially the Science of Things. 'Things' refers to everything in the observable universe - visible and invisible, like objects, sounds, sensations, and ideas. Computers are machines that have been taught to remember the logic of things and their relationships, which is why people say that computers do 'knowledge work.' An electronic computer takes common knowledge about things and arranges the information so that the relationship that exists between units of information can be demonstrated

¹¹Shelley Chasen and Sylvia B. Cohen, **Basic Computing F** (grade 6) (New York, NY: Scholastic Inc., 1984) 6.

¹²Jack L. Roberts, **Understanding Computers** (New York, NY: Scholastic Inc., 1984) 8.

¹³Ellen Richman, **Spotlight on Computer Literacy** (Framingham, Massachusetts: Random House School Division, 1982) iv.

¹⁴Jack L. Roberts, **Scholastic Computing: An Introduction to Computers**, Teacher's Edition (New York, NY: Scholastic Inc., 1984) 4.

immediately or later. ... Additional knowledge about things is discovered in many ways, such as describing processes and proposing new ways in which things could happen. People and computers work together in order to 'understand' the way things are."¹⁵

These definitions are vacuous and reveal no insights about computers. The last definition above degenerates into enthusiastic, essentially meaningless technobabble. Although terminology is the simplest aspect of computer-literacy education, even these most basic definitions of the computer itself reveals the superficiality that characterizes teaching about computer literacy.

At worst, the information that is presented is simply wrong. In one class, a computer operating system was defined incorrectly as a program that "directs the flow of electricity."¹⁶ This was a graduate class in computer literacy, taught for teachers, who went forth armed with fundamental misconceptions about how computers work. (Any mention of operating systems is rare. Only one article I have read about computer literacy suggests that computer-literacy students ought to learn anything about an operating system,¹⁷ and several computer-literacy textbooks make no mention of them at all.)

At best, the abundance of jargon (which derives from the emphasis of computer-literacy courses on the mechanics of computer operation) is difficult to absorb, especially for young children. In a typical computer-literacy class, squeezed into an already full school schedule, a great deal of new terminology must be presented quickly. In one class I observed, when the teacher asked review questions about previous lessons, students responded by stating every new term that had been presented in the previous class.

¹⁵Carin E. Horn and James L. Poirot, **Computer Literacy: Problem-Solving with Computers**, second edition (Austin, Texas: Sterling Swift Publishing Company, 1985) 66, 4.

¹⁶Classroom observation.

¹⁷Wilma S. Longstreet and Peter E. Sorant, "Computer Literacy - Definition?" *Educational Horizons* (Spring 1985) 118.

2.2 Hardware

A common element of computer-literacy education is knowledge of the main components of a computer system. EDUCOM, a consortium of Boston-area schools, stated this in 1983.¹⁸ When secondary-school educators in Nebraska were surveyed to determine which computer-literacy topics they thought most important, they listed ten topics under the themes of awareness about computers and competency; one aspect of awareness about computers was understanding the parts of a computer.¹⁹ Virginia's teacher-certification requirements for computer literacy include identifying the basic components of a computer and describing the functions of each component.²⁰ In 1984, Tennessee's high-school graduation requirements stated that "any student graduating from a school of higher education should be computer literate. Briefly, the terms of this requirement include ... a working knowledge of computer equipment"²¹ Vassar College's core computer-literacy course, "Computing as a Resource," includes instruction designed to make students "understand the basic operating concepts of computers."²²

Typical primary and secondary classes in computer literacy divide the class's hardware segment into input, processing, and output. Most of the time in class is spent on computer input devices and their use. This provides opportunity for a show-and-tell class. I observed one class in which the teacher displayed not only the few components that students would use in the rest of the course (keyboard, terminal, and floppy disks), but also components that are too expensive or that operate on machines that are too expensive for most schools to afford (joystick, mouse, and/or touch-sensitive display). These latter items were brought out from the school's storage closet and passed around the class during the hardware discussion.

¹⁸EDUCOM *Computer Literacy Newsletter*, 1, 1 (Fall 1983).

¹⁹Tina T. Cheng and Dorothy Jo Stevens, "Prioritizing Computer Literacy Topics," *Computers in Education*, 9, 1 (1985) 10.

²⁰Pipho, 101.

²¹M. Gene Bailey and Rebekah L. Tidwell, "Implementing Computer Literacy: The Year After," *ACM SIGCSE Bulletin*, 18, 1 (February 1986) 24. *Proceedings 17th SIGCSE Technical Symposium on Computer Science Education*.

²²EDUCOM *Computer Literacy Newsletter*, 1, 3 (Spring 1984).

This kind of instruction makes it possible for schools that have no computers at all to offer computer-literacy classes. But no sensible computer-literacy class that aims to teach students how to *use* computers can be taught without hands-on access. There is extremely important information about computers that can be taught without hardware or software – for example, one could examine the social and political context of some important computer applications, such as air traffic control, stock-market trading, or military battle management – but these topics are addressed very superficially, if at all, in existing computer-literacy classes.

Little class time is spent on output, perhaps because schools can afford very few printers, very poor-resolution monitors, and no other output devices. Little time is spent on processing, which reflects not an economic problem but a teacher-training problem. The processing phase, while most critical, is poorly explained by textbooks and teacher-training materials. Nothing in this phase of actual computing is visible or tangible. The CPU is simply presented as “the brain of the computer,” and the student is given a mysterious definition of what it does: “This is where the computer solves a problem. You can’t see anything happening in the process step.”²³

Further clarification comes (occasionally) only by dividing the CPU into an arithmetic unit and a control unit. The control unit is not elaborated on further. The arithmetic unit (only occasionally called an arithmetic/logic unit) is often presented as a number cruncher: “The arithmetic unit can only add and compare numbers.”²⁴ It is entirely up to the student to bridge the gap in understanding between a machine that adds and compares numbers and a text processor, for instance.

Course goals that may sound important on paper often are translated in the classroom into mechanical skills. “Knowing a computer’s basic structure and functions” translates to memorizing simple descriptions of computer components, like those mentioned above. “Knowing how to use a computer” translates to learning to handle floppy disks. “Knowledge of computer technology” – one of the three major areas of “computer competence” as defined by the NAEP – translates to knowing some definitions and being able to make

²³Shelley Chasen and Sylvia B. Cohen, *Basic Computing E* (New York, NY: Scholastic Inc., 1984) 8.

²⁴Richman, 24.

the most basic identifications (e.g., pointing out the keyboard in a drawing of a computer system).

In the area of computer-technology knowledge, the NAEP reached the conclusion that "Most American students have some familiarity with computers. Most, for example, can identify a keyboard, disk drive, and printer. Most have some practical knowledge of computers. A majority of seventh and eleventh graders, for example, can distinguish between hardware and software. ... Overall, students were able to answer most of these questions correctly – most had a substantial knowledge of computer functions and components." Their positive tone is not always consistent with the data on which their conclusions are based. For instance, among 11th-grade students (who scored best on the survey):

- approximately 30% did not know what a cursor does,
- 60% did not know what a modem does, and
- 40% could not identify a spreadsheet as a software component or a video display as a hardware component.

To answer the questions correctly, students did not have to know how to define items in any depth; they merely had to pick the correct definitions from multiple-choice lists.

The average performance of 11th-grade students on "knowledge of computer technology" was 65.3% correct. (11th grade students performed best of all the students tested by the NAEP.) As the NAEP concluded, the students answered "most of these questions" correctly.²⁵ By usual standards, they also failed the test.

2.3 Software

Along with an introduction to hardware, some instruction in "basic software concepts" is part of typical computer-literacy classes: "How Computers Work: An introduction to software is a useful exercise for anyone today Removing the mystery of computers will remove some of the exaggerated value now attached to anything that is done

²⁵National Assessment of Educational Progress, 10-14, 29.

by computer.”²⁶ Sometimes the software section of a computer-literacy course is presented as a way to make students feel comfortable interacting with computers. To one elementary-school computer teacher, computer literacy is “just knowing how to approach the machine, what the possible uses are, what kinds of software there are”²⁷ In a survey of educators to determine the most important components of computer literacy, “the competency for public high school students to operate computers ranked the highest on the scale.”²⁸

To the EDUCOM computer consortium, “the ability to use the computer effectively in one’s profession” means emphasizing applications software, not programming or operating skills.²⁹ In Virginia, computer-literacy teachers must be able to “load and run selected software” and “select and use an appropriate program to accomplish a given task.”³⁰ In the classroom, “being able to operate a computer” means being able to run one or a small number of particular pieces of software, described vaguely as whatever software packages people “need.” Often, the goal of “running programs” is satisfied by operating the computer as an appliance, to accomplish a particular goal; this is distinct from knowing anything about how the computer works or how to program it.

The application that dominates computer-literacy classes is word processing. Word processing constitutes *all* of some computer-literacy classes. For example, the only computer requirement for students in Scarsdale, New York, is to know how to use a word processor.³¹ When the New York State Education Department surveyed 14 “people active in computer education,” to get their views on computer literacy, one person classified word processing as “pseudo-programming” and essential to any computer-literacy course.³²

²⁶Jane M. Fritz, “Rethinking Computer Literacy,” in Karen Duncan and Diana Harris, eds., *Computers in Education* (New York: IFIP, 1985) 706. *Proceedings IFIP TC9 4th World Conference on Computers in Education - WCCE 85*.

²⁷Interview.

²⁸Cheng and Stevens, 12.

²⁹EDUCOM *Computer Literacy Newsletter*, 1, 1 (Fall 1983).

³⁰Pipho, 101.

³¹Hearsay, Second National Conference on Technological Literacy (Washington, D.C.: 6-8 February 1987).

³²Center for Learning Technologies, State Education Department, *Computer Literacy: An Introduction*, 2.

In one Massachusetts town, the first of two computer-literacy courses is described as follows: "Computer Literacy I: This course will focus on word processing. Students will learn how to use a word processor, save and retrieve files from a disk and how to print out results in various formats. ... Students will also learn about the different types of software available."³³ Another Massachusetts town divides its computer-education curriculum into two major categories, word processing and, on the same level, "problem-solving and thinking skills." This town has a six-year word-processing curriculum: In the early grades, "informal word processing experiences" means learning to use the keyboard. Later, "introduction to word processing" means typing in text, moving the cursor (one character at a time), and learning two commands (SAVE and RETRIEVE). Finally, "development of word processing skills" and "mastery" each mean learning another few commands.³⁴

Despite the strong emphasis on word processing, 11th-grade students scored only 72% correct on the NAEP's multiple-choice survey questions. This is much better than their performance on the other two commonly taught computer-literacy applications: the same students scored 53% correct on databases and 31% correct on spreadsheets.³⁵ Coverage of spreadsheet programs and database management systems is minimal, focusing on the syntax of a few commands, with little or no time spent on the most difficult aspect of using these systems – organizing information so the programs can be used effectively. The amount of time spent on applications is much less than the typical written curriculum's emphasis on applications suggests (although common-sense suggests that little time for new subjects can be squeezed out of school days):

"The striking pattern of this table is that students seem to be getting infrequent, if any, practice using word processing, graphics, or database applications. Even among those who used the common applications, the frequency of use is low. Only about 10 percent of seventh and eleventh graders prac-

³³Marianne Gonzalez, "Elementary Computer Curriculum Retrospective" (Acton, Massachusetts: Acton Public School System, 12 November 1985) 2.

³⁴"Computer Education Curriculum: Kindergarten-Grade 6" (Arlington, Massachusetts: Arlington Public Schools, September 1985. Draft.

³⁵National Assessment of Educational Progress, 9.

ticed as much as once a week. Graphics and database packages were used less frequently. ...

If [computer science is one of the New Basics], we must be concerned about what appear to be low levels of computer competence described in this report. Students should know more about computers. But what aspects of computer knowledge and use are essential for students to learn? Some argue that, above all, students should know how to use flexible applications software, such as word processing and spreadsheet programs. If these are important, then clearly the data are disappointing. Most students do not have such skills, nor do they have the instruction necessary to develop the skills."³⁶

When other types of software are included in computer-literacy classes, they include spelling checkers, electronic mail, graphics, and telecommunication programs. A suite of software – along with the big three above – is perceived by some people as the new basic training for workers in today's "information society":

"The clear emphasis on total office automation – universal word processing, electronic mail, electronic filing systems, and complex networking – suggests that the corporate world of the future appears headed toward a 'paperless office.' It therefore seems appropriate to include among computer literacy skills the capacity to process text, data, and graphics in a totally electronic environment."³⁷

As I discuss in Chapter 5, arguments for computer literacy that depend on workplace needs are poorly founded.

While inclusion of many types of software makes a computer-literacy course description sound thorough, in class the extra software is likely to be demonstrated by the teacher for a few moments, with little or no time allocated for student use, let alone mastery or significant understanding. Thus, computer-literacy course goals that sound significant on paper – for instance: "The computer-literate student understands what a

³⁶National Assessment of Educational Progress, 21, 70.

³⁷Longstreet and Sorant, 119.

computer can do and has the skills necessary to use a computer to accomplish a variety of tasks"³⁸ – are satisfied in the classes I observed and read about by insignificant "exposure" to a few types of application software.

Although computer-literacy definitions stress general concepts, classes focus on specific software and hardware. The Illinois-Wisconsin computer coordinators' survey is typical in its emphasis on generality:

"... it is vital that computer literacy units stress broad concepts and transferable ideas rather than specifics. It is far more important that the student understands searching and sorting than how to use PFS:File, or looping and branching rather than how to program in a particular language. Above all, the student must be able to use a reference manual or user's guide to discover more about the software or hardware s/he is using."³⁹

But in the classroom, the focus usually is on PFS:File or another database program, not on general concepts of, for instance, searching and sorting.

Students are rarely exposed to actual program documentation, making it difficult for them to learn more about a program at a later time. Instead, they see a reworded subset of command descriptions, often just enough to complete their assignments.

Coverage of all applications is characterized by rote assignments that specify every small step to be completed: which disk to insert when, what commands to type at each point, and so on. Leaving little room for thought or exploration, these are the types of exercises which, it is claimed, fulfill the goal of "equipping students with the skills necessary to enable them to effectively use the computer as a tool both in their education and in their careers."⁴⁰

Software simulations, which have the potential to be among the most creative educational uses of computers, are often beyond the financial reach of schools. Some educational programs are simulations of activities that are much better experienced directly.

³⁸"Mission: Define Computer Literacy. The Illinois-Wisconsin Computer Coordinators' Committee on Computer Literacy Report," 10.

³⁹Ibid.

⁴⁰Bailey and Tidwell, 24-25.

One such program attempts to simulate growing a plant: The student specifies an amount of water (in milliliters, a ridiculous measurement for watering plants), number of hours of sunlight per day, and so on. The program determines how much the plant would grow each week and updates a crude stick-figure plant on the screen. Students are asked to repeat this process 14 times, to simulate a 14-week growth cycle; they quickly grow bored with the program. If the students grew plants, they would learn much more about plants, and the education would be much more engaging.

Simulations are removed from reality. They do not include the qualitative texture of the real world or manipulation of real things. Because of the avoidance of direct experience that is necessarily part of simulations, one educational psychologist questions the value of them for young children:

"It is difficult to think of one simulation activity that could be more meaningful to young children than their own dramatic play. ... My consistent emphasis on direct experience is not meant to imply that young children using computers are not experiencing. They are. It is the quality of that experience and the accompanying collateral learnings that are being questioned. What may be added to what has already been noted is that analysis, definition, and logic do not suit every realm of knowing or state of being. Ambiguity, uncertainty, and the qualitative also are a part of knowing and living. It would be ironic if in introducing the microcomputer to young children – a tool with enormous potential for expanding capacity and learning possibilities – we end up limiting the range and quality of their experiencing."⁴¹

A former teacher, now a producer of well regarded software simulations for the educational market, cautions teachers to use simulations judiciously and never forget their limitations:

"A simulation may approach reality, but it is not reality, and never can be. It should never be used if the real thing is available and viable as a learning experience under the conditions one is working. ... But if you settle for

⁴¹Harriet K. Cuffaro, "Microcomputers in Education: Why Is Earlier Better?" *Teachers College Record*, 85, 4 (Summer 1984) 564-565.

simulations because such field trips are out of range or impossible, be aware that much of the richness and color ... cannot be simulated. Most of the variables cannot be known and if they were could not be included. ... The harder developers work to make simulations rich enough to be worthwhile, ..., the harder they are going to be *pedagogically* to use. Why? Because simulations beg for intense teacher involvement, and the better the simulation, the greater the requisite involvement."⁴²

In some computer-literacy course descriptions, the software component is described as a way to remove the students' sense of mystery about computers, but classes often do the opposite. In one graduate class, students became fascinated with the computer's ability to sort a list of words alphabetically, and one student asked the teacher how the computer could do this. Shrugging her shoulders, the teacher responded: "How does it alphabetize? I don't know. How does it know to be a word processor? It's magic!"⁴³

2.4 Programming

A major unsettled curriculum issue of computer-literacy education is whether programming should be taught. Some educators feel that programming is too technical for an introductory computer class. They argue that programming is the domain of computer experts, not people who are merely computer literate.

On the other hand, many computer-literacy classes do include programming. A series of computer workbooks for grades 1-6 focuses on programming (in BASIC and Logo), to the exclusion of applications software.⁴⁴ In 1983, EDUCOM found that most definitions of computer literacy included "the ability to read/write simple programs; BASIC is frequently mentioned as the programming language required."⁴⁵ In 1985, results of an-

⁴²Snyder and Palmer, 1986, 122, 142.

⁴³Classroom observation.

⁴⁴See Patricia Relf Hanavan, **Basic Computing A** and **Basic Computing B** (grades 1 and 2); Patricia Relf Hanavan, Shelley Chasen, Sylvia B. Cohen, **Basic Computing C** (grade 3); and Shelley Chasen and Sylvia B. Cohen, **Basic Computing D**, **Basic Computing E**, and **Basic Computing F** (grades 4, 5, and 6). New York, NY: Scholastic Inc., 1984.

⁴⁵EDUCOM *Computer Literacy Newsletter*, 1, 1 (Fall 1983).

other survey listed several programming-related items among the ten computer-literacy topics appearing most frequently in the literature: "being able to write algorithms or draw flowcharts when using computers to solve problems, knowing programming language commands, and being able to write computer programs."⁴⁶ Acton, Massachusetts's computer-literacy handbook (1983) lists two "essential areas," one of which is computer programming.⁴⁷ Virginia's certified computer-literacy teachers must be able to "write a simple program" to solve a given problem.⁴⁸ In Tennessee, high-school graduates are required to have "a modest ability to use a programming language."⁴⁹ In two New York City guides to computer literacy, most of the curriculum modules focus on programming. The guide for intermediate and secondary grades contains 40 lessons, 27 of which are about BASIC. The guide for elementary grades contains 53 modules on Logo and 23 on "Computer Literacy and the BASIC Language" (most of which are about BASIC).⁵⁰

A variety of arguments are used to justify the inclusion of programming in a computer-literacy course. Some people claim that learning to program makes one better understand the powers and limitations of computing. One Oregon school district has a mandated 6-week, 30-hour, middle-school computer course that "familiarizes students with ... beginning programming skills." The district's computer coordinator explains why programming is included: "You can be more forgiving of the computer's limits and more understanding about why it's doing what it's doing, or failing to do, if you have some level - and it doesn't have to be advanced - of programming competence."⁵¹ The coordinator may have added the caveat about not needing "advanced" programming competence because his district used tiny computers (8 kilobytes of memory), too small for students to acquire genuine programming competence, let alone understand the powers or limits of the much larger computers that run important real-world applications.

Other people believe that a major goal of computer-literacy education is to enable

⁴⁶Cheng and Stevens, 10.

⁴⁷Marianne Gonzalez, 1.

⁴⁸Pipho, 101.

⁴⁹Bailey and Tidwell, 24.

⁵⁰**Computer Literacy: Elementary Grades and Computer Literacy: Intermediate and Secondary Grades** (Brooklyn, New York: Board of Education, City of New York, 1983).

⁵¹David Hawley, **Computer Literacy** (Oregon: Oregon School Study Council, 1984) 14-16. Printed as *OSSC Bulletin*, 27, 8 (April 1984). Quoting Jack Turner.

people to *control* computers – John Kemeny talks about learning to “make them [computers] do what you want them to do”⁵² – and that programming instruction is necessary to achieve this control. But the programming instruction I know of in existing computer-literacy classes is far too superficial to enable students to write meaningful programs or otherwise exert meaningful control over computers.

Still others claim that teaching programming is a means to teach other, more important skills, typically logical thinking skills:

“In computer literacy, we don’t really teach programming as such. We introduce them to programming concepts. We’re teaching them structure, we’re teaching them logical thinking”⁵³

“... if programming and algorithms do not become part of the learning package for developing computer literacy, the learner is deprived of the opportunity to develop, in the context of computer hardware, appropriate cognitive structures for problem solving.”⁵⁴

This argument for computer-literacy education is reminiscent of earlier arguments, for instance, arguments for universal education in Latin. I call this the mental-discipline argument for programming and discuss it (and the lack of justification for it) in Chapter 5.

When pressed, many computer-literacy teachers who include a programming component in their courses do not know why they are teaching programming. Often under pressure to construct a computer-literacy curriculum quickly and with little support, teachers must fill class time by “exposing” students to whatever software can be acquired most easily.

Where programming is part of a computer-literacy course at the primary- or secondary-school level, it means programming in Logo (a language used only in schools) or BASIC (a language widely denounced by computer professionals). Frequently, BASIC

⁵²Roberts, *Scholastic Computing*, 41. Interview with John Kemeny.

⁵³Hawley, 10. Quoting Jim Neideigh, district computer-education specialist.

⁵⁴Bikkar S. Randhawa and Dennis Hunt, “Computers and Computer Literacy in Contemporary Psychological, Socio-Economic, and Educational Context,” *AEDS Journal*, 17, 3 (Spring 1984) 3.

is included gratis with the small computers that are most often bought or donated for school use. BASIC is so often part of the educational-computing package that one textbook refers to it as being "built into the machine" and informs readers that, if they are to communicate with a computer at all, they must learn BASIC.⁵⁵ A computer-literacy activity book for elementary and middle-school students states that "Most microcomputers have a BASIC translator built into their circuitry – that is, BASIC is included in the price of the machine."⁵⁶ Since schools struggle to afford computers, the incentive to use free software is great.

Logo also is often included with a basic educational computer package. Moreover, the education literature has been flooded with enthusiastic articles about Logo. Since computer-literacy teachers do not have the time (and often not the necessary skills) to evaluate the mountain of "education" software, the incentive to use software that everyone else is using is great. There is no time to question the fact that the alleged benefits of Logo have not been well evaluated:

"Most authors of articles on Logo and its uses perceive little need to justify Logo at all; it has been accepted by educators already. Indeed, Logo seems to have become something of a cult within education. D.D. Thornburg, author of seven books on Logo, states that many converts 'accepted (Logo's) benefits on faith and felt that criticizing Logo ... bordered on sacrilege.' Although Logo is now some twenty years old, there has been relatively little empirical research conducted to either refute or support the claims of Papert and his disciples."⁵⁷

Whatever the justification for teaching programming, the classroom experiences I know about are dismal. As with applications software, students' exposure to programming is minimal, if for no other reason than the lack or poor quality of equipment

⁵⁵Stephen Radin and Harold M. Greenberg, *Computer Literacy for School Administrators & Supervisors* (Lexington, MA: Lexington Books, 1983) 59.

⁵⁶International Council for Computers in Education, *Computer Literacy Activities for Elementary & Middle School Students* (Oregon: ICCE, 1984) 23.

⁵⁷Wesley A. Many, et al., "The Effect of Learning to Program in Logo on Reasoning Skills of Junior High School Students," *Journal of Educational Computing Research*, 4, 2 (1988) 204.

(discussed, along with other classroom problems, in Chapter 3). Lack of teacher training, often aggravated by bad teaching materials, is another problem. One textbook talks about "controlling the computer with quality programming,"⁵⁸ but typical computer-literacy teachers are not proficient programmers and are never taught about "quality" programming concepts such as structured programming.

Although documentation is critically important in real-world programming, it is often absent from programming instruction in computer-literacy classes. In one test of the cognitive effects of programming, students were asked to choose the "function of program documentation" from the following:

- "To store the program on a disk safely
- To ensure program success
- To write down codes using a programming language"⁵⁹

Such a trivial reference to documentation in a test reflects classroom instruction that typically makes no mention of, for instance, the relationship between documentation and program maintainability or comprehensibility.

It is senseless even to talk about "quality" programming in most classes, since classroom exposure to programs is so superficial. A course description might include "learning what computer programming is all about. The main goal should be to acquaint students with a first-hand knowledge of what computer languages is [sic] and what it means to write programs."⁶⁰ A typical course focuses on the syntax of a few programming-language commands and the ability to read and write the simplest of programs.

One particularly impressive definition of computer literacy includes this:

"An indirect but important set of skills for computer literacy is the evaluation of computer-stored data and the models used in its manipulation. Evaluation includes the following:

⁵⁸Radin and Greenberg, 6.

⁵⁹Tina T. Cheng, Barbara Plake, and Dorothy Jo Stevens, "A Validation Study of the Computer Literacy Examination: Cognitive Aspect," *AEDS Journal*, 18, 3 (Spring 1985) 150.

⁶⁰Geoffrey Akst, "Computer Literacy: An Interview with Dr. Michael Hoban." *Journal of Developmental Education*, 8, 2 (1985) 17. Quoting M. Hoban.

- identifying the premises underlying a data structure,
- determine the appropriateness of program operations, and
- assessing the validity and reliability of the outcomes.”⁶¹

Concepts such as data structure or program reliability are not mentioned in any of the computer-literacy classes, curricula, and textbooks I studied. What can a student learn about high-quality programming by being exposed to 10-line BASIC programs? Moreover, students are not made aware of just how merely introductory their programming exposure is.

The NAEP reported distinctly negative results in its survey of students’ understanding of computer programming:

“As a whole, students knew little about programming in the languages most commonly taught in schools It can be argued that only small numbers of students have experience with programming languages. But the picture is changed only moderately when observations are confined to students who said they know the languages. ... Most students appear to have had difficulty answering the assessment questions. No grade averaged even 50 percent correct Overall performance appears to be disappointing.”⁶²

In a typical NAEP survey question, 69% of 11th graders could not select the correct definition for algorithm from the following multiple-choice list:

- A step-by-step process for solving a given type of problem.
- A word processing program for the computer language ALGOL.
- A special procedure for interpreting computer output.
- A special program for algebra.⁶³

⁶¹Longstreet and Sorant, 118.

⁶²National Assessment of Educational Progress, 25, 28.

⁶³Ibid., 13.

This is not surprising to anyone who observes computer-literacy classes, in which "algorithm" is considered a sophisticated concept and is rarely discussed.

A more typical computer-literacy test question, which reflects course content, asks students to match a list of programming-language names with application areas; e.g., "COBOL" is correctly matched with "business." Another test question lists a five-line BASIC program consisting entirely of PRINT statements, and asks students to make a series of changes; e.g., add a line that causes the computer to print a particular phrase.⁶⁴ Both questions are intended for students who have completed a programming component of a computer-literacy course.

In one survey of programming instruction, the Bank Street College of Education interviewed students who studied Logo. The students had significant misconceptions about the concept of a program, what happens inside the computer, recursive programming, saving and recalling programs, and the concept of an algorithm. The researchers concluded: "the progress of even the most advanced students could be hindered by their inadequate mental models of Logo. ... the children we studied had neither deep knowledge of Logo nor explicit problem-solving strategies. ..." ⁶⁵

Another study, of high-school students who were taught programming, concluded this: "All measures of programming skill showed that the students had gained only a modest understanding of any of the languages taught." Most students did not understand flow of control, order of execution, recursion, values of variables that are incremented, and even some basic programming-language commands. Few students used variables or subprocedures; instead, most students wrote linear, brute-force programs or were unable to solve assignments at all. Few students could accurately predict the output of a program that had an iterative structure with conditional tests and multiple variables to be incremented.⁶⁶

⁶⁴"Assessments for Computer Literacy," Dayton, Ohio: Dayton Public School System (1986) 1-2.

⁶⁵Ronald Mawby, et al., "Structured Interviews on Children's Conceptions of Computers," Technical Report No. 19, Center for Children and Technology, Bank Street College of Education (February 1984) 36.

⁶⁶D. Midian Kurland, et al., "A Study of the Development of Programming Ability and Thinking Skills in High School Students," *Journal of Educational Computing Research*, 2, 4 (1986) 40-43.

2.5 Jobs in Computing

Many computer-literacy courses discuss "careers in computing." The Salem, Oregon school district offers computer literacy in all its middle and high schools. Its course goals include: "Be aware of career opportunities related to computers. Be aware of current uses of computers."⁶⁷

The usual motivation for telling students about jobs in computing is a belief that these students are entering a world in which most, if not all jobs will involve computers. Much of the time in computer-literacy classes is spent on what may be viewed as vocational education, concentrating on the mechanics of operating machines, running application programs, and learning jargon. But the job market in computing is overrated by computer-literacy textbooks, which imply that any computer experience whatsoever will assure students of good jobs:

"Schools need to teach students the skills that are fast becoming passports to better-paying jobs. These skills include word processing techniques, calculating routines, alternative ways of transmitting data, and the use of electronic mail and electronic bulletin boards."⁶⁸

The only jobs for which these skills might be considered particularly important are low-paying clerical positions. As I discuss in Chapter 5, few jobs in which people *use* computers require people to know how anything *about* computers.

2.6 Social Impacts of Computers

"Social impacts" is a catch-all phrase that encompasses computer uses, computer ethics, and legal implications of computers. One education-journal article about computer literacy argues for a substantial social-impacts component in computer-literacy courses: "To

⁶⁷Hawley, 18.

⁶⁸William J. Ridley and McAllister, H. Hull, Jr., "Transforming American Education: Reducing the Risk to the Nation. A Report to the Secretary of Education, United States Department of Education, by the National Task Force on Educational Technology" (Washington, D.C.: U.S. Department of Education, Office of Educational Research and Improvement, April 1986) 15.

be literate means to understand how computers and other aspects of modern technology affect and interact with every aspect of our life: in school, on the job, in the home and at play. Students should understand how to cope with the changing technology and how to interact with the many applications of computers around us.”⁶⁹ But, as the article’s author complains, most computer-literacy classes do not impart understanding of social impacts of computing, because most class time is spent on the mechanics of operating the machines.

Many course descriptions include general topics such as “how computers are changing our lives,” “proper and improper uses of computers,” “what’s good about them – and what’s dangerous,” or “an awareness of the public debate over how computers may or should affect our social institutions and our individual lives.”⁷⁰ In the classroom, “social effects of computers” means a brief mention of the fact that some people lose their jobs because of computer automation. “Computer ethics” means a brief mention of computer crime. Ethics discussions in computer-literacy classes include nothing about whistleblowing, the largest category in the literature about computer ethics and an area of active discussion in the computer community. “Legal aspects of computers” means a little more than a warning to students that they should not make their own copies of software disks they use in class.

A sixth-grade computer-literacy workbook has a section on “People and Computers.” It consists of four cartoons with a list of captions below each cartoon. The student is asked to select the most appropriate caption from each list. The correct answers are:

1. “Many people believe that computers can do everything, but computers cannot work without people.
2. Computers are not only for playing games.
3. The word *bug* has a special meaning in computer language.
4. Some people think that computers want the same things as humans.”

⁶⁹Alan Neibauer, “The Computer Literacy Myth,” *T.H.E. Journal* (February 1985) 89.

⁷⁰Roberts, *Scholastic Computing*, 1, 41 (interview with John Kemeny).

According to the "Skills Chart" at the back of the book, this exercise is the *entire* coverage for the following "skill": "Recognizing social and other non-technical implications of using computers."⁷¹

Classroom and textbook coverage of the impacts of computers is quite optimistic. Computer-literacy classes ignore or treat only lightly the weaknesses and negative impacts of computer applications. For instance, students are told that the FBI uses computers to store information on crimes and criminals. They are not told that surveys of the record quality of these files revealed that as many as 72% of some files' records were inaccurate, incomplete, and/or ambiguous.⁷² Students are told that the Department of Health and Human Services uses computers to maintain information on senior citizens eligible for Social Security benefits, but they are not aware that the huge system is so incomprehensible that SSA programmers are afraid to try to correct problems by modifying the software.⁷³ Students are told about allegedly high-productivity uses of computers in factories, but they do not discuss job losses due to computer automation – except to be assured that they need not be troubled by such job losses, because laid-off factory employees can simply get new jobs repairing computerized equipment.⁷⁴ Such teaching cheerfully ignores evidence that factory workers' skills are distinct from computer-repair skills, retraining programs are rare, and some companies install computerized equipment explicitly to reduce the size of their labor force, hence have more interest in firing employees than in transferring them to other positions. Computer-literacy classes do *not* consider uses of computers that are among the most ethically problematic for computer professionals – military uses, such as in autonomous weapons.

⁷¹Chasen and Cohen, **Basic Computing F**, 62-63.

⁷²See Kenneth C. Laudon, **Dossier Society: Value Choices in the Design of National Information Systems** (New York: Columbia University Press, 1986). See also U.S. Congress, Office of Technology Assessment, "Federal Government Information Technology: Electronic Record Systems and Individual Privacy," OTA-CIT-296 (Washington, D.C.: U.S. Government Printing Office, June 1986).

⁷³Hearsay, SSA project at Computer Corporation of America.

⁷⁴Roberts, **Understanding Computers**, 59.

2.7 Powers and Limits of Computers

Underlying educators' enthusiasm about computer-literacy classes is the belief that learning about computers will enable students to control the computers in their lives and to be wise citizens in an "information age." Computer-literacy skills, it is argued, will help equip today's students to be tomorrow's technological decision-makers. According to this argument, only a computer-literate population will be able to intelligently evaluate computer uses, identify problems and potential problems, and resist dangerous and otherwise undesirable new systems. I believe that education about computers that emphasizes informed evaluation of computer uses *is* important; but this sort of "computer literacy" would require an accurate and profound understanding of the powers and limits of computers, issues that are covered at best superficially in typical computer-literacy classes.

Often, the level of explanation is simplistic to the point of inaccuracy. Genuine understanding of the powers of computers is stymied by the oversimplified models of computing put forth in computer-literacy classes. One textbook teaches that computers just work with numbers.⁷⁵ This entirely misses the point that computers derive most of their power from their ability to process abstract symbols. Most computers at work in the real world spend little time "crunching" numbers.

On one computer-literacy test, the correct answer to the question "How does a computer solve a problem?" was "It follows instructions to do what it is programmed to do."⁷⁶ A textbook states: "A computer can't do anything by itself. It must be told exactly what to do and how to do it."⁷⁷ And in classes I observed, this is exactly what students are taught: computers do (only) what they are told to do. This is comforting but, as every computer professional knows, it is dangerously misleading; more on this in Chapter 6.

Genuine understanding of the limits of computers is also stymied by superficial classroom coverage. One list of objectives for computer-literacy education includes this: "Lim-

⁷⁵Jack L. Roberts, *Scholastic Computing*, 5.

⁷⁶Cheng, Flake, and Stevens, 148.

⁷⁷Roberts, *Scholastic Computing*, 10.

itations: ... focus on developing a general sense of the capabilities and limitations of computing machines. Examples of computer limitations include the facts that computers do not have feelings and consciousness, nor are they able to make value judgements. ..."⁷⁸ But while computers cannot "make" value judgments, computer systems embody the value judgements of their designers. To understand the limitations of computer simulations, for example, one must know that they are abstractions of reality that are "loaded with judgements about what to include and what to leave out."⁷⁹

A computer-literacy curriculum guide states that after taking the course described in the guide, "Students will be able to discuss the limitations of the computer – such as its inability to synthesize information, and originate new ideas."⁸⁰ But computer systems can combine and manipulate information and can interact with other systems in ways that are novel and entirely unexpected, even to the people who programmed those systems. This behavior is characteristic of most large computer applications. People who are unaware of it, its sources and consequences, do not understand one of the most basic limitations of computer use.

Inaccurate and superficial classroom materials nurture inaccurate and superficial models of computing. When the Bank Street College of Education studied one computer-rich school, they found that students developed important misconceptions about computing. The students had incorrect models of the inner workings of computers and the powers and limits of computers, and they "knew nothing" of the fundamental notion of the computer as an abstract computational device.⁸¹ Another study reached a similar conclusion:

"Most students do not understand the general limitations of computers. For instance, more than half of the 1,500 students tested believe that 'computers help people make decisions by providing correct answers to any question.'

While those with no prior exposure to computers were more likely to give the

⁷⁸Ronald E. Anderson and Daniel L. Klassen, "A Conceptual Framework for Developing Computer Literacy Instruction," *AEDS Journal*, 4, 3 (Spring 1981) 144.

⁷⁹Snyder and Palmer, 1986, 122.

⁸⁰Center for Learning Technologies, State Education Department, **Computer Literacy: An Introduction** 77. Quoting Pat Walkington, "Curriculum Guide for Beginning Computer Literacy and Programming in BASIC," Phoenix, Arizona.

⁸¹Ronald Mawby, et al.

incorrect response to this item, 43% of the students with computer programming experience also answered incorrectly. ... It is clear ... that students lack realistic views and basic knowledge of computers and that mere exposure to computers does not automatically eradicate misconceptions.”⁸²

Development of an informed critical attitude about computers is undermined by some schools' explicit determination to “teach” students to think positively about computers. In a list of the ten most important computer-literacy topics, Nebraska secondary-school educators included “having positive attitudes about computers.”⁸³ A wider survey of educators' views about computer literacy, which generated a prioritized list of topics, placed “having positive attitudes” in the upper half of the list.⁸⁴ One article included the following objectives for computer-literacy classes:

- “*Enjoys and desires* work or play with computers, especially computer-assisted learning.
- *Describes* past experiences with computers with positive-affect words, like fun, exciting, challenging, and so on.
- Given an opportunity, *spends* some free time using a computer.”⁸⁵

As early as third grade, the NAEP found that more students said they liked using computers than had ever used computers!⁸⁶ Students may be influenced not only by positively biased curricula but also by positively biased media reporting about computers (not to mention computer-company advertisements).

Most disturbing is the dangerously misleading statement in all computer-literacy education, that computers just follow instructions:

“Computers can only do what people tell them to do in a program. If a computer does not get clear instructions, it won't do anything. If something

⁸²Anderson and Klassen, 141-142.

⁸³Cheng and Stevens, 10.

⁸⁴Ibid., 12.

⁸⁵David C. Johnson, et al., “Computer Literacy – What Is It?” *Mathematics Teacher*, 73, 2 (February 1980) 94.

⁸⁶National Assessment of Educational Progress, 32-33.

is wrong in a program, the computer will tell you by saying ERROR. Then you must correct the mistake.”⁸⁷

Many of the large programs on which we depend are so complex that no one understands how the system of which they are a part functions. It is this more subtle level of explanation that is necessary for informed, critical decision-making about the uses and abuses of computers.

Computer-literacy education primarily teaches recognition – of jargon, components, and applications – on very simple computers, and at a superficial depth of mere multiple-choice “awareness.” It produces students who believe themselves to be “computer literate” but who know nothing about real-world computing. Recognition is not the same as understanding.

⁸⁷Shelley Chasen and Sylvia B. Cohen, *Basic Computing E*, 60.

Chapter 3

Educators' Perspectives

Most of the teachers I interviewed recited a litany of problems with computers in their schools. Many complain that there is not enough equipment. The hardware that their schools can afford is at the inexpensive end of the personal-computer spectrum; these are small machines that are relatively hard to use, with constraints that can affect educational objectives (e.g., some of the word-processing programs that run on small machines can manipulate only very short papers). There is universal agreement that the quality of most "educational" software is abysmal.¹

Many teachers are frustrated by the lack of training they receive in the use of computers. In one school, teacher training consisted of one hour of instruction given by the local Radio Shack store, provided gratis with the purchase of Radio Shack computers for the school. Some teachers who are required to teach computer literacy are given no training whatsoever and feel that they get insufficient ongoing support. Some districts have mandated computer-literacy courses and no computer-literacy curricula, mandated courses and no computer coordinator, or mandated courses and no funds allocated for computers. Even people who specialize as computer coordinators are trained primarily in fields other than computing. Few people with significant computer training consider working under the conditions that characterize primary- and secondary-school teaching.

Other problems exist in the support environment that is needed with computers; too

¹Information presented in this Chapter is based largely on interviews and classroom observation conducted in several school districts in Massachusetts. My interview and observation data are supported by many references in the education literature that discuss education nationwide.

little funding is available for hardware maintenance and software review, not to mention continued investment in new hardware (computers become obsolete *very fast*). Schools may not fully appreciate the enormous extent of these support costs. It is not in the interest of computer salespeople to emphasize these costs.

Computer-literacy curriculum development also gets short shrift. There are districts that buy computers but have no idea how they will use them. There are districts that require computer-literacy courses but have neither course curricula nor the resources to develop such curricula. One district created a new position, computer coordinator, but it still has no computer curriculum; their computer coordinator wrote that in a report:

"Curriculum Development: A written core curriculum does not exist. Thus, the use of computer technology as a tool for instruction is not identified. Staff Development: A core academic curriculum does not exist. Thus, the existing staff development model for computer education is not driven by the core academic goals for the City. ... There currently exists no written document that outlines what practices are going on within the system as it relates to the use of computers to achieve curriculum goals."²

Although many schools have installed computers, a two-year survey of 2,500 schools found no evidence of substantive curriculum changes because of computers.³

Despite its shortcomings, the present environment of school computing has been achieved only at great cost relative to school budgets. Funding for computers seldom comes from new sources, but from grants and school budgets that are already severely strained. In 1983, money spent on software and computer hardware in schools totaled one-third of all money spent on books in all subject areas and all grades.⁴ Time is another hard-pressed scarce resource, as computer literacy or any other new topic cannot be added to the curriculum without either extending class time or taking time away from other topics.

²Krepelka, "Action Plan," 11, 16.

³Robert Pearlman, Educational Computer Operations, System Specialist, Boston Public Schools. "Computers and Education," talk given in a series on Computers and Society, Massachusetts Institute of Technology, 19 March 1985.

⁴Henry Jay Becker, "Computers in Schools Today: Some Basic Considerations," *American Journal of Education*, 93, 1 (November 1984) 22.

School systems rarely consider such critical tradeoffs of money and time explicitly. The standard operating procedure for schools has been to buy computers before thinking about software and to buy both hardware and software before thinking about teacher preparation. Curriculum development comes later still. Schools rarely stop to consider *whether* computers can contribute meaningfully to educational programs; if they do consider this, it is usually only after making significant investments in computer instruction.

There is no denying the fact that many educators are enthusiastic about having computers in schools. Compared to the normally slow, staid progress of the educational system, the introduction of computers has been a cultural explosion. Multiple streams of influence contribute to teachers' motivations to use computers; they often have little to do with education. Indeed, many teachers who are enthusiastic about computer-literacy education do *not* place computer literacy near the top of their lists of educational priorities. To begin to understand their motivations, one must consider computer literacy in the broader context of the educational system as viewed through teachers' eyes. For instance, teachers cannot help but be impressed by the prestige conveyed by computer use in schools, even where more basic facilities are lacking. The single most prominent theme to emerge from my discussions with teachers about computer literacy is their sense of being *pressured* to use computers. This and other major concerns of classroom educators about computer-literacy education are discussed below.

3.1 Hardware and Software

One of the most surprising aspects of the strong rhetoric in favor of computer-literacy education is that so many people have spent so much time discussing the impact of a machine to which students and teachers alike have so little access. Teachers who use or want to use computers in their classrooms complain chronically about the lack of hardware. They are right — financial investments in hardware that are large by school standards brought the typical school (K-12) to a ratio of only 1 computer for roughly every 40 students, as of April 1986, when the last large-scale survey was done.⁵ The

⁵Ridley and Hull, 11.

same ratio was reported in a large survey done in the spring of 1985. The 1985 survey also reported the following figures:

- 86% of U.S. schools had at least one computer for instruction, 63% had 5 or more, and 24% had 15 or more. The median was 8 computers per school.
- The median percentage of students who used computers in an average week was 30% in K-6, 25% in middle schools, and 21% in high schools.
- The median amount of computer use per student per week was 35 minutes in K-6, 1.5 hours in middle school, and 1.8 hours in high school.
- The most active 10% of students accounted for 33% of the time that computers are used in schools. The most active 25% of students accounted for 60% of the computer use.
- The median number of hours per week that school computers were in use was 15 in K-6, 20 in middle schools, and 23 in high schools.⁶

The problem of lack of hardware is compounded by the inadequacy of the hardware that schools typically can afford to buy. The small computers that students are typically exposed to in school are extremely limited in what they can do; for instance, in the length of document that they enable students to process. Graphics or sound, where available, are crude. These and other inadequacies produce what one prominent computer scientist calls "kindergarten-level training" in computers.⁷

Poor facilities for computer-literacy education affect teachers' lives in ways not necessarily directly related to computer-literacy education. The scarcity of computers creates a logistical nightmare for the school administration and particularly for already overburdened teachers who are required to teach computer literacy. School computers are either hoarded in one room or dispersed sparsely throughout individual classrooms. If they are

⁶Becker, "Instructional Uses of School Computers" (June 1986) 2, 5, 6.

⁷Daniel D. McCracken, "A Skeptical View of Computer Literacy," *SIGCSE Bulletin*, 16, 1 (February 1984) Appendix A, *Proceedings 15th ACM SIGCSE Technical Symposium on Computer Science Education*, 5.

gathered together in a separate computer lab, teachers must figure out how to fit lab time into the school week. When lab time is a particularly scarce resource, teachers sign up for it in advance, then arrange the rest of their schedule around the computer lab.

Few computer labs contain as many machines as there are students in a typical class, so students must either share machines or use them on a rotating basis. Shared use may have the potential for interesting collaborations among groups of students, but thus far, there is little software or curriculum materials that tap this potential. In addition, successful supervision of collaborative work is especially demanding on the teacher. Rotating use is the norm when computers are dispersed throughout classrooms, typically giving each teacher one or two machines. This can lead to serious classroom-management problems: teachers may feel that they need to spend an inordinate amount of time "directing traffic." The even smaller number of printers also affects class plans; teachers may have to set aside enough class time for everyone to line up and print hardcopy versions of their work before class ends, or students may leave their computer classes with no written product to review.⁸

The outdated machines that characterize school computer labs can run only limited software; that is, only software which runs on extremely small machines. Much of the recently well reviewed educational software (e.g., complicated simulations or environments for exploring mathematics, science, and other subjects) needs more storage capacity than that found in a typical school computer. Such software runs on machines that are much too expensive for schools to afford; often, the software itself is too expensive for schools to afford. The hardware limitations of school computers rule out, for instance, the use of contemporary, icon-based, WYSIWYG ("what you see is what you get") word-processing software. Other examples are spreadsheet and database-management software designed to run on typical school computers. Many teachers and students are left to contend with word-processing (or other) programs that are hard to learn and to use and that have long since been replaced by better software in, for instance, much of the commercial workplace that does word processing.

⁸Classroom observation.

When programming (as opposed to applications software) is included in computer-literacy classes, it can be done only in relatively unsupportive software environments:

"Good programming environments ... make ... coding ... far more efficient, allowing the programmer to concentrate on higher level issues In contrast, the programming environments provided for today's school microcomputers are so impoverished ... that entering the code for a program and getting it to execute correctly is the central problem."⁹

Software designers producing programs for the pre-college educational market must seriously compromise their programs (e.g., in terms of ease of use, as manifested in the quality of user interfaces) because of limitations of the typical school computers on which those programs will run. In short, schools attempting to plan for computer use often find that issues unrelated to educational needs and goals severely constrain what they can do with computers at all, let alone educationally.

The vast majority of "educational" software (including software designated for computer-literacy classes and other subjects) is widely denounced by educators as not satisfying their curriculum needs. Teachers I have seen reviewing a variety of school software have criticized programs because of instructions that were confusing, incomplete, or written at the wrong grade level, because of poor pedagogy, poor choice of colors, and inadequate documentation. Few educational software programs are field tested before distribution.¹⁰ One paper, about a district that is proud of its computer-literacy program, says this: "In at least one regard, South Brunswick is just like every other school district that uses computers. Everyone there agrees that the available educational software is, for the most part, terrible."¹¹

⁹Roy D. Pea and D. Midian Kurland, "On the Cognitive Effects of Learning Computer Programming: A Critical Look" (New York: Bank Street College of Education, Center for Children and Technology, January 1984), Technical Report No. 9, 10.

¹⁰Curt Dudley-Marling and Ronald D. Owston, "The State of Educational Software: A Criterion-Based Evaluation," *Educational Technology*, XXVII, 3 (March 1987) 25.

¹¹Dennis A. Williams, "How One School Does It Right," Article 20 in Terence R. Cannings and Stephen W. Brown, eds., *The Information Age Classroom: Using the Computer as a Tool* (Irvine, California: Franklin, Beedle & Associates, 1986) 96.

Software-evaluation organizations rate the majority of commercially available educational software as just plain bad. According to the Educational Products Information Exchange (EPIE), only 5% of the software they evaluated as of 1985 was "exemplary," and about 25% did not meet either their minimum standards or standards set by the California State Department of Education in its "Guidelines for Educational Software Evaluation for California Schools."¹² Another software-evaluation service, MicroSIFT, "highly recommended" only 17% of the software it evaluated (up to 1985).¹³ In 1984, a Canadian software project in the province of Alberta was unable to recommend nine out of ten software products reviewed for use in schools.¹⁴

Teachers' and educational organizations' claims of poor quality software are supported by a study in which computer-literacy tests were given to more than 3,000 students at 3 grade levels in 115 schools. Test questions were based on instructional objectives from computer-education curricula nationwide. Students who were trained with computer-literacy software averaged only 50% correct in all grades.¹⁵

Some software publishers are book publishers who know little about software but are pressured to include it in their educational materials. Many of them turn existing textbooks into software, producing workbook-like programs known as "drill-and-practice." So much "educational" software is of the drill-and-practice variety that *most* of the computer time in elementary schools is devoted to drill-and-practice programs.¹⁶ While this type of software is easy for teachers to insert into an existing curriculum and to use, many teachers (and students) quickly grow bored with it.

"On the whole, skill-oriented software is very similar to workbooks in content, with the exception that these 'workbooks' move. ... The question here is not 'Why use a microcomputer?' but rather 'Why use workbooks, animated or not, with young children?'"¹⁷

¹²Dudley-Marling and Owston, 26.

¹³*Ibid.*

¹⁴*Ibid.*

¹⁵Roy M. Gabriel, "Computer Literacy Assessment and Validation: Empirical Relationships at Both Student and School Levels," *Journal of Educational Computing Research*, 1, 1 (1985) 419.

¹⁶Art Levine, "Something doesn't quite compute," *U.S. News & World Report*, 101, 19 (10 November 1986) 78.

¹⁷Cuffaro, 562.

That is, we should not use computers to bring out the worst we have to offer in educational materials. For their focus on uninspired software, Judah Schwartz, Director of Harvard's Educational Technology Center, feels that most software publishers are "beneath contempt."¹⁸

The hardware and software problems that characterize computer-literacy education are unlikely to disappear soon. With few exceptions, the educational-software market is responding to the uncritical demand for software by quickly mass-producing huge quantities of low-quality programs. Research and development of higher quality software, designed to satisfy teachers' curriculum needs, are rare. The U.S. school system is not in the business of funding software development. "Computers and education" work in the computer-science community is focused mainly on sophisticated simulations and microworlds (e.g., to help teach physics) that are not relevant to computer-literacy programs and that run on machines that are more powerful and expensive than typical school computers. It is unclear how schools could finance the acquisition of more expensive machines.

As scarce as computers are in schools, there are many more computers than ancillary equipment and supplies needed for computer-literacy instruction. I visited an affluent suburban private school that had a full-time computer coordinator, but because of a shortage of software, disks had to be shared among students.¹⁹ Because a typical school computer lab has only one or two printers, teachers may have to end class early, to give students enough time to line up and print their work. If the teacher decides that class time is more important than printing time, the students may not be able to print anything; instead, they must wait until their next computer class to review and continue their work.

These are the situations that have spawned computer-literacy textbooks and curricula designed to teach the use of computers, but with little or no actual use of computers: "Even with the most generous budget, schools introducing computers for student courses must plan for a rotating use of each terminal. In other words, the average CL course

¹⁸Judah Schwartz, "Software to Think With," talk presented at CPSR/Boston meeting (Computer Professionals for Social Responsibility) 2/17/88.

¹⁹Classroom observation.

has a built-in premise: at least part of it must be taught without a computer."²⁰ In one computer-literacy activity book, 11 of the 12 activities do not involve the use of a computer; the one activity that does require a computer calls for a total of 15-30 minutes of hands-on access.²¹

To understand the role of even the most limited computer facilities in schools, it helps to know something about overall school facilities and resources:

- Many primary- and secondary-school teachers work in environments that are quite literally decaying physically. A report on urban schools, recently issued by the Institute for Educational Leadership, found that the backdrop for teaching is often leaking roofs, burned-out lights, and broken toilets.²² The findings of this report are consistent with other national and state surveys. In 1983, there was a \$25 billion backlog of school repair and renovation work; today, maintenance in many schools is emergency-driven at best.²³ Maintenance is such a serious problem that, according to teachers, if physical conditions are to be even tolerable, the school's principal must take an active role in selecting and monitoring the school's custodial staff.
- Studies of schools have noted persistent, serious shortages of basic pedagogical and other necessary resources. Many schools do not have enough textbooks, reading kits, desks, blackboards, dictionaries, science equipment, lab workbooks, typewriters, and even pencils and paper. The Institute for Educational Leadership reported that "In *most* schools, there was a 'rationing atmosphere' about basic supplies"²⁴ (my emphasis). Of 31 schools surveyed, 25 reported less than adequate resources, citing insufficient supplies of materials, equipment, and staff. In some cases, teachers and principals are driven to buy their own supplies, ranging from blackboards to toilet paper, using their own money.

²⁰Roberts, *Scholastic Computing*, 3.

²¹*Computer Literacy Activities for Elementary & Middle School Students*, 10.

²²Thomas B. Corcoran, Lisa J. Walker, and J. Lynne White, "Working in Urban Schools" (Washington, D.C.: Institute for Educational Leadership, 1988) 11.

²³*Ibid.*

²⁴*Ibid.*, 28.

- Space is also in short supply: Some teachers conduct classes in auditoriums or other large spaces, where classroom areas are demarcated by movable partitions. In these "rooms," noise from classes on the other side of the partitions is clearly audible and obviously disruptive. Some teachers conduct classes in closets or in small lounges, where they are disturbed by people passing through on their way to bathrooms.²⁵ "Floating" teachers, with no classroom of their own, are common.
- Teachers have limited access to basic office equipment, including copy machines, telephones, audio-visual equipment, and typewriters. Copying facilities are a particular problem; copy machines are often considered a luxury, and access to them is severely limited. In some cases, only administrative personnel may use a school's copy machine, and teachers are relegated to ancient mimeograph machines. The Institute for Educational Leadership's report noted a school that had exceptionally good resources. In that school, teachers had purchased their own copier, financed with the proceeds of candy sales they held on their own time throughout the year. The teachers continued to hold these sales to obtain funds for supplies and maintenance, even after the machine was purchased.

I met several teachers who were thrilled about the prospect of having a computer at their school, not because they felt it could contribute to their educational goals, but because they could use it to print their lesson plans and would no longer have to use "ditto" machines. They had a critical need for office equipment and, in this context, it made sense for them to view the computer as a copier. I had not even seen a ditto machine in many years, and I was surprised initially that none of the teachers suggested addressing the office-equipment problem directly, by purchasing a copier. The teachers knew that parents and administrators were likely to approve funding for computers but not for copiers.

To summarize, many primary- and secondary-school teachers work in environments that they consider professionally unsupportive and personally demoralizing. Working under such conditions, some teachers do not expect to be successful, whether they are

²⁵Ibid., 15-16.

teaching computer literacy or anything else. When computers are introduced in these environments, often along with much fanfare, some teachers react with enthusiasm. What is not clear is the extent to which their enthusiasm has anything to do with the educational value of computers. Teachers' reactions to school purchases of computers may more strongly reflect what is bad about schools than what is good about computers.

3.2 Funding

When school districts, state education departments, and so on, decide that their schools are to teach computer literacy, they are not always forthcoming with funding to support the endeavor at the individual school level. Funding is needed, on a continuing basis, for curriculum development, teacher preparation, hardware and software acquisition, and maintenance. According to a survey by the Education Commission of the States, most school officials in states with mandated computer-literacy instruction said that no "new or specific money" was appropriated for the purchase of computers in local school districts.²⁶ I encountered districts with some requirement for computer-literacy instruction but no resources for curriculum development or teacher training.

From 1981 to 1986, Federal funding to the 44 largest school districts dropped 20%, and local sources could not provide full replacement revenues.²⁷ In spite of the climate of strained educational budgets, schools have spent significant amounts of money on computing. Schools have financed computer acquisitions in a variety of ways, most commonly by tapping large portions of discretionary funds, including state and federal block grants. Sometimes, funds used to buy computers and software are taken directly out of other budget areas, such as for books and other materials, or for building repair and maintenance. In California, one computer-rich class called the SmartClassroom bought \$110,000 worth of electronic equipment; to finance the purchases, "the district used its share of state lottery proceeds, normally earmarked for teacher salary raises

²⁶Pipho, 100. Citing "Minimum High School Graduation Course Requirements in the States," *Clearinghouse Notes*, November 1984, Education Commission of the States.

²⁷Corcoran, Walker, and White, 5-6.

...."²⁸ Parental interest in educational computing has played a large role in school-computer funding: one study showed that about one-quarter of the computers bought for elementary and secondary education in 1983 were paid for with funds raised by parents explicitly for that purpose.²⁹

Despite expenditures for computers that are large by school standards, the computing environments in typical primary and secondary schools are poor. Cost is a deciding factor in the quality of a school's computer-literacy program. At East Tennessee State University, software that integrates word processing, spreadsheets, and other applications was considered preferable pedagogically – preferable to different programs for different applications, each with its own user interface, documentation, and so on – but the integrated software package was too expensive.³⁰ In a class on "Computers and Writing" (for teachers), the teachers agreed that computerized spelling programs are extremely inconvenient to use on machines with only one disk drive; they also agreed that most of their schools could not afford to buy two disk drives per machine."³¹ In one computer-literacy class, even the small cost of floppy disks was an issue; the teacher told students that they could avoid buying extra disks – backup disks – by getting a hardcopy printout after each session.³² Another computer-literacy class used a particular word-processing program not because it was deemed the best such program (or even a good one), but because the teacher was able to buy 10 copies of the program (a "lab pack") at a reduced cost. Usually, teachers can afford only one or two copies of a program, which must then be shared among a whole class.³³ One computer coordinator commented that when hardware or software is purchased by a school, not because it is good or fits well into the curriculum, but because of cost, teachers must adapt to the computers, rather than vice-versa.

²⁸David S. Wilson, "Computer's Role in Class Expands," *The New York Times* (13 April 1988) B:11.

²⁹Marc S. Tucker, "Computers in the Schools: What Revolution?" Article 21 in Terence R. Cunnings and Stephen W. Brown, eds., *The Information Age Classroom: Using the Computer as a Tool* (Irvine, California: Franklin, Beedle & Associates, 1986) 107.

³⁰Bailey and Tidwell, 25.

³¹Ronni L. Rosenberg, "Computers and Writing," class paper for Ethnographic Research in Educational and Community Settings, Harvard University (May 1985) 32.

³²Classroom observation.

³³Classroom observation.

Moreover, despite the best of intentions on the part of educators to provide equal access to all students, access to computers is unequal. In 1983, "80% of the 2,000 largest, richest high schools used computers for instruction, while only 40% of the smaller, poorer high schools did."³⁴ Similar inequities continue to be documented by more recent studies. Schools located in high socio-economic areas have more computers than schools located in low socio-economic areas. Schools with a racially mixed or predominantly minority student body have fewer computers than other schools.³⁵ These gaps grow larger from middle school to high school. The same surveys indicate that that the most academically successful students receive the most personalized instruction with computers. With too few computers to satisfy all the teachers who want or have to use the machines, some schools must make hard choices about priorities; when one typing teacher wanted to use the school's computers for word processing, she was told that "computers are only for the academically talented."³⁶ In this respect, allocation of computing resources is consistent with other areas of spending for education. Such inequities may be systematic, since part of school budgets comes from local taxes, and funding for computer acquisition sometimes comes directly from local parents.

3.3 Training: Little Change from Kindergarten through Graduate School

All the computer-literacy teachers I spoke with were trained as teachers in areas other than computing.³⁷ Teachers bear most of the burden of implementing computer-literacy education, but usually they feel they do not have adequate opportunities to prepare to teach computer-literacy courses. Often, teachers with little or no preparation find themselves required to teach computer-literacy courses. One woman was put in charge of introducing computers into a Boston-area school district while she was taking an

³⁴Tucker, 1986, 107. Quoting Market Data Retrieval press release, October 1983.

³⁵Becker, "Instructional Uses of School Computers" (August 1986) 11.

³⁶Neibauer, 88.

³⁷Barbara Doyle, "Should a Computer Literacy Class be the First Required Course for Data Processing Majors?" *SIGCSE Bulletin*, 18, 2 (June 1986) 11 [ACM Special Interest Group on Computer Science Education.] See also National Assessment of Educational Progress, op cit.

introductory computer-literacy course.

Whatever their motivations for learning about computers, many teachers feel they get little support for their efforts. In-service computer training is often considered inadequate. Many teachers make heroic efforts to educate themselves with whatever bits and pieces of information they can gather from friends and colleagues. Some take college courses in computer literacy or other types of educational computing. These efforts are made after a work day that, for an average teacher, might include teaching a half-dozen classes, with a total of 150 students, and taking papers home to grade and lesson plans to prepare at night. Under such circumstances, teachers are lucky if they can stay a small step ahead of their students.

Teacher training in computers is minimal in terms of the time devoted to it. Typically, when such training is available at all, it consists of only a few hours of workshops, during which teachers primarily listen and have little opportunity to use computers or see them in use over a long time. In one school, only one instructor with "special training" was allowed to use the computer equipment; the training consisted entirely of a nine-hour course at a local computer store.³⁸ A computer-literacy activity book for elementary and middle-school classes states that a half-day or one-day workshop is sufficient preparation for teaching a computer-literacy course, for people who have had no prior experience with computers.³⁹

The computer training that teachers do receive is user training. Training classes are designed to impart just enough mechanical knowledge about how to operate the machine so as not to be overwhelmed by it, not any understanding of how computers work. Training may be done by computer-store representatives who know nothing about educator's needs or training methods. "Most of the present computer workshops for educators ... are conducted by specialists with only marginal knowledge of the complexities of elementary education, and there is little connection between the course content and the real world of the classroom."⁴⁰

In computer-literacy training classes, computer knowledge is trivialized. In one com-

³⁸Neibauer, 88.

³⁹International Council for Computers in Education, 9.

⁴⁰Langdon Winner, *The Whale and the Reactor*, Chicago: University of Press (1986) 155.

puter workshop for elementary teachers, "the presenter emphasized that 'You don't need to know anything about how this thing works. Just put the disk in, press RETURN, and if it doesn't work, try another disk.'" ⁴¹ An Apple II tutorial disk claims to be an introduction to the machine but covers only "keyboarding"; at the end, it prints a message on the screen: "Congratulations, you're now a true-blue computer expert." ⁴² One computer coordinator said she considered it a *strength* of her district that there were no "specialized" computer teachers; in this environment, she viewed everyone as a computer teacher. ⁴³

Consider these statements from classroom materials:

"You already know a great deal about computers. You have gained knowledge of computers through reading, watching television, going to movies and talking to people. Computers have been readily available and widely used in our society for about twenty years. Just by living in this society, you have learned a lot about computers You may not be consciously aware of some of your computer knowledge. For example, our telephone system is heavily computerized. ... Many of the newspapers and magazines you read are produced using computerized typesetting equipment. Computers may print your paycheck and send withholding data to the Internal Revenue Service." ⁴⁴

"... students are entering college with considerable computer experience. ... This student has probably spent several thousand hours playing video games, is very familiar with the state of the art computer graphics [sic] as a result." ⁴⁵

But people whose interactions with computers are limited to receiving computerized bills, reading computer-typeset publications, or playing computer games are computer "users" only incidentally, and learn nothing *about* computers through these interactions. In the absence of adequate training in computers, computer-literacy teachers lack the skills to

⁴¹Calfee, 11.

⁴²Classroom observation.

⁴³Dennis Williams, 96.

⁴⁴International Council for Computers in Education, 6.

⁴⁵Klassen, 43.

evaluate the materials provided to them, and they face a frightening challenge when the system does not work as predicted.

Sometimes, the information about computers that is presented to teachers is misleading, confusing, or simply wrong. In one class, I witnessed the following discussion of operating systems:

"An operating system is a program that tells the computer how to deal with information – tells it how to move information, how to operate, how to do things. ... An operating system is done in a lower level language, machine language. It really controls the flow of electricity through the circuits."⁴⁶

Later, when discussing Logo, the teacher explained that "Logo is like an operating system, another set of instructions to the computer."⁴⁷ Videodisks were described as a "sort of database, sort of an educational medium."⁴⁸ When one of the students became confused about how a videodisk is like a database, the teacher said that the term "database" was used loosely, "to mean it is a collection of information of the same kind, and you can access each one separately."⁴⁹

Observations I made of teachers learning programming are very similar to observations others have reported of children learning programming. I cited several such studies in Chapter 2. Another study, of a class that had 45 hours of intensive Logo instruction (much more programming instruction than a typical computer-literacy class), noted these findings:

- Programming instruction emphasized syntax, not models of how programming languages work. Students "seemed to understand the code in that they gave adequate glosses of individual lines. But ... many revealed that they did not understand Logo's control structure well enough to trace the program's execution. ... Even more failed to understand the most basic fact of flow of control: after a called

⁴⁶Classroom observation.

⁴⁷Ibid.

⁴⁸Ibid.

⁴⁹Ibid.

procedure is executed, control returns to the next line of the calling procedure.”⁵⁰

- Students often generated effects by trial and error, without any underlying mastery of the language. “One characteristic of highly interactive programming languages such as Logo and BASIC is that students can often get the effects they want simply by repeated trial and error, without any overall plan, without fully understanding how effects are created, without the use of sophisticated programming techniques, and without recognizing that a more planful program could be used as a building block in future programs.”⁵¹
- “Brute force” programming was common, showing little evidence of students engaging in problem decomposition or using language features to structure a solution to a problem. “For example, if a similar shape is required several times in a program, students will write new code each time the effect is required, rather than writing one general procedure and calling it repeatedly. Programs thus consist of long strings of Logo primitives that are nearly impossible, even for the students who have written them, to read, modify or debug.”⁵²

Introductory programming instruction, at the level found in typical computer-literacy courses,⁵³ emphasizes syntax and superficial mechanics *at the expense of* correct mental models and deep understanding of programming concepts. This emphasis characterizes teacher training in programming (as part of computer-literacy training), and teachers propagate their own training when they teach computer literacy.

Overall, teachers and even computer coordinators rate their computer training as poor. The National Assessment of Educational Progress found that “Many school [computer] coordinators have minimal training in computer studies and rate themselves as

⁵⁰D. Midian Kurland, et al., “Mapping the Cognitive Demands of Learning to Program,” Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (New Orleans, Louisiana: American Educational Research Association) 21.

⁵¹Ibid., 4.

⁵²Ibid., 3.

⁵³I refer here to programming instruction in typical computer-literacy classes, not to higher level programming courses. I do not refer, for example, to programming-concepts courses that might be introductory courses in a multi-year computer-science curriculum.

mediocre in their ability to use computers."⁵⁴ Of teachers surveyed in grades 3 and 7, less than one-quarter rated themselves very good at using computers; one-third rated themselves fair or poor and did not consider themselves adequately prepared. One-third of 11th-grade teachers rated themselves very good at using computers, and 15% rated themselves fair or poor.⁵⁵ In another survey, the percentage of "computer-using" teachers who considered themselves expert at various computer tasks was 10% in grades K-6, 21% in middle school, and 27% in high school.⁵⁶ And teachers are not supported by other, better trained staff; in schools that I visited or read about, there was no one with significant training in computers.

Curricula are effected by the level of teacher training: people cannot teach what they themselves know little about. Computer-literacy teachers naturally propagate their own computer-literacy training. Consequently, in many computer-literacy classes, directions specify every minute step of the process of operating computers; e.g., which disk to insert when, what disk drive to use, what to type at each step, when to press Carriage Return, what to name each new file, and so on. When studying programming, sample programs are handed out and students may be asked to "Type the following exactly as it is written."⁵⁷ In typical computer-literacy classes, satisfying a course goal of "learning to use the computer" more than likely means following a sequence of directions of what to type when.

Mastery of new skills is a strong motivator that can contribute to increased job satisfaction. But most teachers feel they have less than adequate professional-development opportunities.⁵⁸ While teachers reason that their students will be computer literate if they learn as much about computers as they themselves know, they feel that their own computer-literacy training is insufficient. *Indeed, it is astonishing how little the depth of computer-literacy education changes as the grade level increases from kindergarten through graduate school!*

⁵⁴National Assessment of Educational Progress, 3.

⁵⁵Ibid., 64-69.

⁵⁶Becker, "Instructional Uses of School Computers" (June 1986) 4.

⁵⁷Classroom observation.

⁵⁸Institute for Educational Leadership, 79.

Criticism of teacher-training programs in a variety of areas is not new; it has come from both outside and inside the education community. In "A Nation at Risk," the National Commission on Excellence in Education found that "teacher preparation programs need substantial improvement. ... The teacher preparation curriculum is weighted heavily with courses in 'educational methods' at the expense of courses in subjects to be taught."⁵⁹ In part because time and funding for teacher training is limited, the character and distribution of training in computer literacy raises general questions about educational training priorities. I have seen teachers who do not know basic math making time for computer-literacy training. Should computer training be a high priority for teachers who do not know or remember that there are 360 degrees in a circle, or that there is a relationship between the number of sides in a (regular) polygon and its interior angles?⁶⁰ For a prospective teacher who cannot correctly add and subtract negative numbers, how important is computer-literacy training?⁶¹ What priority should it be given?

3.4 Priorities

Some computer-literacy enthusiasts include computer literacy in their short lists of educational basics they feel all students must learn:

"All we can hope to give our children to allow them to function in this complex world are the fundamentals upon which they can continue to learn. The fundamentals include:

- Reading and comprehension skills
- Communication skills (writing and public speaking)
- Mathematical concepts and problem-solving
- Fundamentals of scientific principles
- Computer skills"⁶²

⁵⁹National Commission on Excellence in Education, 22.

⁶⁰Classroom observation.

⁶¹Ibid.

⁶²Fritz, 709.

But normally, a new topic cannot simply be added to the existing curriculum. Unless the school day or school year is lengthened, a new topic can be added only by replacing something. In their "Sample Elementary Computer Curriculum Objectives," EDCO (the Educational Collaborative, in Massachusetts) puts it this way: "Since any new course takes *time* from existing curricula, systems must reorder their priorities."⁶³ One question for computer-literacy advocates is this: What is computer literacy more important *than*?

Answering this question requires considering overall educational priorities. Many students who graduate from high school have appalling gaps in their knowledge in areas usually considered part of basic education. The National Assessment for Educational Progress was recently (1988) commissioned by ABC News to do a large, national survey of high-school students. Some results were reported in an ABC News special report on American education:

- More than half the surveyed students did not know what apartheid is.
- Nearly 70% did not know what Chernobyl is.
- Two-thirds did not know who the vice-presidential candidates were (in the last few months of the 1988 Presidential campaign).

The average correct scores in the tested areas were: history 54%, literature 52%, math 63%, and science 55%.⁶⁴ All failing scores, by normal standards. Similarly distressing results are reported in many other studies.

One ought to critically question how important computer literacy is, in the light of such problems. A secondary-school teacher expressed her view of the proper place for computer education among overall education priorities:

"There's so much in the news now about functional illiteracy, etc. Well, we're teaching the kids computers, we must be doing something. I have nothing against teaching the kids computers in school. I think if you have a big high school, with a big faculty, and you have the time to give them a minicomputer

⁶³"EDCO Sample Elementary Computer Curriculum Objectives" (Newton, Massachusetts: 1985) 1.

⁶⁴**Turning Questions. America's Kids: Why They Flunk.** ABC News special program (3 October 1988). Transcripts: Journal Graphics, Inc., 267 Broadway, New York, NY 10007.

course, I say fine. But the thing I want to know is, can those kids add and subtract without the computer? ... Can they read and write? Can they write a good sentence? If they have to do a job resume, are they going to be able to write it? Can they balance their checkbooks? If you can do that and still fit in the computers, I say great. ... I really feel that if you don't even ever show a kid a computer, but he can read well, if he can think and has good math skills, he can pick up a computer manual any time he wants and figure out how to use it. There are so many gaps. Right now, our primary goal is to teach the kids how to think for themselves, and how to read and how to write, and to be coherent, and to enjoy learning and be able to function in life."⁶⁵

Some computer-literacy advocates argue that time for computer-literacy instruction need not come at the expense of other instruction, if computer literacy is *integrated* into the existing curriculum, rather than taught as a separate course. They contrast more traditional, *standalone* computer-literacy instruction, in which computers are treated as objects of instruction, with *integrated* computer use, in which computers are treated as educational tools for instruction in subjects other than computer literacy. Integrated computer use is sometimes referred to as computer awareness, computer exploration, or computer specialization; none is well defined.

Integrating computers into a school's existing curriculum requires much more curriculum-development effort than creating a separate computer-literacy course: Integrated computer use requires knowledge of not only computers but also the subject into which computers are being integrated. And while a separate computer-literacy course requires one curriculum-development effort (or perhaps one effort for each level of instruction), integrated computer use requires curriculum-development efforts for every course in which computers are to be used. As I have shown above, resources even for development of separate computer-literacy courses are insufficient. Hence, for practical reasons, the integrated approach to computer education often means taking the compo-

⁶⁵Interview.

nents of an existing computer-literacy course and inserting them as is in several existing courses. For instance, word processing might be covered in English class; databases, in history class; spreadsheets, in math class; and "keyboarding," in primary school.

Integration of computers into existing curricula places a premium on sufficient quantities of equipment, well targeted software, and well prepared teachers. In the absence of all these items – that is, in typical schools – the integrated approach to computer literacy has not often succeeded; the National Assessment for Educational Progress found that computers are not used much in teaching subjects other than computer-literacy.⁶⁶ And like the standalone approach to computer literacy, the integrated approach still requires a share of scarce educational time and money for hardware and software review and acquisition, curriculum development, and especially teacher preparation.

The call to make computer-literacy education a high priority is based much less on demonstrated benefits than on enthusiasts' claims for *potential* benefits:

"Computer learning environments have the potential for imparting some ... important higher cognitive skills. ... Programming instruction has the potential for fostering the higher cognitive skills called for by the many recent reports on the state of educational practice but, so far, the potential is not being achieved."⁶⁷

Representative Albert Gore, Jr., author of the National Educational Software Act of 1984, has said this: "The potential for computers to improve education is enormous, more dramatic than any invention since writing. Yet that potential is not being met [sic]. Simply put, our schools are being swept up in a tidal wave of technology without any idea of how to make wise use of it."⁶⁸ Often, technical reasons are given to explain why computers have not (yet) lived up to their potential; for example, that there is not enough hardware or that existing software is bad. After many years of unfulfilled

⁶⁶National Assessment for Educational Progress, 35.

⁶⁷John Dalby, Francoise Tourniaire, and Marcia C. Linn, "Making Programming Instruction Cognitively Demanding: An Intervention Study," Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association) (New Orleans, Louisiana, 9-10) 2, 19.

⁶⁸Albert Gore, Jr. Quoted in Bonner, 21.

promises about educational computing, some educators are beginning to react negatively to the wide gap separating rhetoric from reality:

"Software that is supposed to turn the world around is disappointing us right and left, and *we're not paying attention*. ... As policy-makers we still insist that everything is fine and that we just need more money, more time, and more expertise."⁶⁹

The doubt felt privately by "every teacher and parent" is ignored collectively.⁷⁰

While educational computing is founded on *potential*, teachers are painfully aware that the most problematic *realities* of education are unlikely to be ameliorated by technology. In 1987, a Carnegie Foundation study found that 25% of urban teachers and 13% of other teachers said undernourishment is a serious problem for schools.⁷¹ "You can't teach someone who is hungry, or cold, or worried about being shot, worried about having a child, or considering suicide."⁷² Preoccupied with survival, many families cannot make education a priority. Teachers also cite poor student discipline and disruptive behavior as major problems in schools, often linked to a lack of parental support for education. Three out of every four parents never visit their child's school, never get to know their child's teacher.⁷³

"People want their kids to go through school. And they want them to get A's on their report cards. But they don't necessarily want them to learn anything or to do any supervision that's necessary at home. ... Somehow they think we have some magic wand that we can wave over their heads, and they're just going to sit up and be smart, do all their work, and not have to bring any of it home. And of course, we run into the same thing that any other teacher would; we have a lot of parents who were not taught right themselves and

⁶⁹Snyder and Palmer, 18.

⁷⁰Ibid.

⁷¹Corcoran, Walker, and White, 137. Referencing the Carnegie Foundation for the Advancement of Teaching. **An imperiled generation: Saving urban schools**. Princeton, New Jersey: The Carnegie Foundation for the Advancement of Teaching, 1988.

⁷²**Burning Questions. America's Kids: Why They Flunk.**

⁷³Ibid.

who can't help their kids. We have two 1st-graders who probably read as well if not better than their parents. You can tell the parents who went to school, because you can see the difference in their kids. The kids in kindergarten, you can tell the kids whose parents read to them and those who just spend a lot of time playing or watching TV."⁷⁴

And students spend plenty of time "playing and watching TV": half of all students watch three to five hours of television every day, while only one-third spend as much as one hour per day on homework. One out of every four students drops out of school before high-school graduation.⁷⁵ In this environment, how high a priority is computer-literacy education?

3.5 Pressures to Use Computers

The single theme encountered most often in my discussions with teachers about computers is their strong sense of being pressured to use computers in class. The pressure comes from several sources that interact with each other to form a complex web of social forces.

For instance, there is a commercial aspect to computer literacy: the developers, manufacturers, and retailers of computer-literacy materials (materials for classroom use and for teacher training) have an economic incentive to encourage the education establishment to buy into computer-literacy education. "We're looking at an infinite market," said the chairman of one computer-manufacturing firm.⁷⁶ Computer-literacy textbooks send powerful messages to teachers about the dangers of being computer illiterate: "Before long, if you don't have your own computer, ..., you'll be competing with people who have more information and skills than you do."⁷⁷ The television advertisements of major educational-software vendors send a clear message that computers in schools are good, even necessary, for students.

As we have seen, teachers almost unanimously complain about the poor quality of

⁷⁴Interview.

⁷⁵**Burning Questions. America's Kids: Why They Flunk.**

⁷⁶Douglas Noble, "The Underside of Computer Literacy," *Raritan* (Spring 1984) 61.

⁷⁷Roberts, *Scholastic Computing*, 15.

school computing materials which, they say, do not address their curriculum needs and appear to have been created by people who know nothing about classroom realities. But high quality software, including educational software, is difficult, time-consuming, and expensive to produce and field-test. Because teachers' own computer-literacy education consists of only minimal (if any) exposure to tiny programs and no discussion of characteristics of larger programs, they are not prepared to fully appreciate the difficulties of large-scale software production. Because schools feel the need to use computers, without having the resources to evaluate "educational" software packages on the market, teachers buy software in spite of their distaste for it. The uncritical software market so created fuels the industry, which has filled catalogues with hundreds of pieces of software for the educational market. Indeed, in a sense, the segment of the computer industry that produces materials for schools is simply responsive to pressure from educators for more computer-literacy materials (and other computer software and textbooks). Some of the resulting programs, produced hastily, are little more than new, dull, computerized versions of old, dull, workbook drills.

Teachers also talk about feeling pressured to use computers by their students' parents: "Today, parents are gauging the value of their child's education by the number of computers in the classroom."⁷⁸ One computer coordinator explained:

"A lot of interest? Oh, a lot of it is – you open a new computer lab, the PTA comes in, they'll have an open house, and the parents certainly marvel at the fact that there's all sorts of equipment. People feel good about that because it's something concrete. It's much more difficult to say how valuable is that computer lab going to be in that child's education."⁷⁹

On the other hand, parents, like teachers, are responding to a bombardment of newspaper and magazine stories about the omnipresence of computers and the alleged need for everyone to know about computers (see Chapter 5 for more discussion of the reasons given for teaching computer literacy). And parents also see the television advertisements of educational-software vendors, some of which explicitly link being a good parent to

⁷⁸Neibauer, 89.

⁷⁹Interview.

buying a computer. Neither parents nor teachers are exposed to much critical evaluation of the claims for and the results of computer-literacy education.

One of the most obvious sources of pressure on teachers to use computers comes from education administrators who impose requirements for computer-literacy instruction. Teachers usually have little involvement in decision-making about school requirements. Some teachers feel that school administrators "get the computers first, then apply pressure on teachers to use it, instead of identifying an educational need and asking if a computer could help with that."⁸⁰ One kindergarten teacher lamented:

"I don't think these kids are developmentally ready for computers and have trouble with the fact that computers are there. They are used as worksheets, which is bad. Kids need concrete manipulation, they're not yet abstract thinkers. They're learning some things, but in spite of the computers. ... But I have to use the computers, so I tried to make a checklist of things to look for in software. I'm having trouble because I don't think they should be used at all. The back to basics movement in schools means drill and practice, and this is just transferred to computers, although it is already developmentally wrong. I do have a computer in my room and don't know what to do with it."⁸¹

David Noble suggests that by placing heavy emphasis on the introduction of computer technology and the "need" of students to be computer literacy, administrators "stand to expand their realm School administrators are laying their bets on a sure ticket to a better life for schools – technology."⁸² But administrators respond similarly to many of the same pressures to use computers that teachers and parents feel:

"They [principals] naturally want their school to have the most modern educational tools and curriculum. Today, the measure of this seems to be the number of computer terminals in the classroom and the number of students enrolled in literacy classes. Because of the speed by which computers have

⁸⁰Classroom observation.

⁸¹Interview.

⁸²Noble, 61.

grown in our society, the placement of hardware precedes the training of teachers, the construction of curriculum and even the ability of teachers.”⁸³

Finally, many teachers talk about feeling strong pressure to use computers from “society in general”: “You feel pressure from the fact that you hear so much about computers.”⁸⁴ They are referring to a society-wide fascination with computers, one manifestation of which is the massive media attention to the “computer revolution.” Teachers absorb a strong message that today’s students *must* know *something* about computers, but there is little substantive discussion of why this is or is not so, or what sort of information about computers is appropriate or inappropriate.

Teachers are sensitive to public pressure on them to teach about computers, but many remain unconvinced of the educational value of computer-literacy instruction. As a result, many teachers feel “pushed into it [using computers]; they feel that they ‘have to learn’ about computers”:⁸⁵

“The computers *are* coming, whether we like it or not.”⁸⁶

“This is a unique phenomenon in the history of education, where the will of the public for an untried and expensive new subject is placing it in many curricula before the educators are either convinced about its place in education or prepared to handle the subject completely and in a coordinated fashion. Public approval is forthcoming for the expenditure of large sums of money to put microcomputers in school at a time when money is hard to come by, the quality of our public school education is being seriously challenged, and teachers are being laid off.”⁸⁷

Even some of the teachers who are extremely enthusiastic about using computers reflect strong external pressures when they talk about why they use computers:

⁸³Neibauer, 89.

⁸⁴Interview.

⁸⁵Ibid.

⁸⁶Naomi S. Baron, “Should Everyone Learn Everything: The Question of Computer Literacy,” *SIGCSE Bulletin*, 16, 1 (February 1982) 114. *Proceedings 15th SIGCSE Technical Symposium on Computer Science Education*.

⁸⁷Fritz, 705.

"It's going to happen. Computers in the school has got to happen, and it's got to happen at all levels. ... The computer is not a hula hoop to get stuck in the closet. John Wayne isn't going to show up and ask, 'Are they bothering you, Miss? Well, I'm back and I'll take care of it.' So, you may not have to use it in your class, but it's going to happen around you."⁸⁸

To some extent, the call for computer-literacy education feeds on teachers' (and the public's) fears of emerging somehow disadvantaged from the "computer revolution":

"People who don't use computers in the future will be different, left behind."⁸⁹

"You feel pressure from the fact that more kids are turning in papers done on computers, and that the kids know things that you don't know."⁹⁰

"They're here to stay, and you're going to have to know how to use them. I don't want my students to be smarter than me."⁹¹

Some teachers also feel the need to have early, active involvement in school computing, to help ensure teacher *control* over the new curriculum:

"Educators from all disciplines must learn as much as possible so that they can wrest control from the software publishers, the hardware manufacturers, the college professors and the anxious parents operating out of fear rather than educational vision. Who should be pushing the buttons? You should."⁹²

To better understand teachers' sensitivity to being "left behind," to losing control, and to pressures related to how they do their job, it helps to understand more about the pressures that normally dominate their jobs. All too often, primary- and secondary-school education in the U.S. is characterized by a lack of prestige for teachers. Most teachers (especially in cities) do not feel respected by society, parents, district administration, and supervisors:

⁸⁸Interview.

⁸⁹Ibid. 13.

⁹⁰Interview.

⁹¹Ibid.

⁹²Steve Bergen and Lynne Schalman, "Who's Pushing the Buttons?" *Classroom Computer Learning* (December 1984) 55.

"Teachers are burned out, to a large degree. Teachers feel people don't respect us, because of the way we're paid, because of the conditions, and because we're not treated like professionals, we're treated like very tall children."⁹³

A frequent complaint from teachers is that their supervisors treat them like students; e.g., teachers at one school were insulted when their principal gave awards to teachers for perfect attendance.⁹⁴ Their jobs are further trivialized when they are required to spend much of their time doing things that have little to do with their profession; a typical teacher's duties might include several of the following tasks: hall monitoring, lunch duty, bus supervision, study hall, cleaning, smoking duty, late night duty, or counseling tardy students.

In 1983, the report "A Nation at Risk" cited several critical problems with teaching: teachers were not well enough prepared, their working life was "unacceptable," salaries were too low, few of the best students were attracted to teaching, and teachers had little influence on professional decisions.⁹⁵ In 1988, another report on teaching cited many of the same problems and noted that many teachers feel "powerless" to change the conditions of their working lives: most have little control over school priorities, staff development, school regulations, budget development, and resource allocation.⁹⁶

To a large extent, teachers work in isolation in their classrooms (that is, when they have their own classrooms); they have relatively few opportunities, time, or space for professional contact with their colleagues. Rewards for teaching are seen as inadequate, and sanctions for bad teaching are much more in evidence than rewards for good teaching. Today, half of all new teachers choose to leave the profession within five years.⁹⁷ Many of the teachers who stay have concluded "We don't count."⁹⁸

Because teachers have little control over important aspects of their jobs, it is natural that some of them are anxious to jump on the computer-literacy bandwagon early enough

⁹³**Burning Questions. America's Kids: Why They Flunk.** Quoting Mary Futrell, President, National Education Association.

⁹⁴Corcoran, Walker, and White, 84.

⁹⁵National Commission on Excellence in Education, 22-3.

⁹⁶Corcoran, Walker, and White, 61.

⁹⁷Susan Tift, "Who's Teaching Our Children?" *TIME*, 132, 20 (14 November 1988) 62.

⁹⁸Corcoran, Walker, and White, 64.

to exert control over the new technology. Because they feel beleaguered by a society that is extremely critical about education, it is natural that some of them are eager to respond quickly and positively, if uncritically, to societal enthusiasm about computers. Because they work in physically deteriorating environments with a scarcity of basic resources, it is natural that some of them want to take advantage of the one resource that administrators, parents' groups, and legislators are willing to buy for schools. In many cases, enthusiasm about computers and education has nothing to do with the educational value of computers.

3.6 Technology-Driven Education: 'A Bandaid on What's Wrong with the School System'

One reflection of the pressure-driven nature of school computerization is the fascination among computer-literacy enthusiasts with technology for technology's sake, not for educational value. Standards for educational programs are often technology-driven, "determined by the latest introduction from the commercial hardware manufacturers – more memory, faster speed, better graphics, improved sound output, more attractive appearance or type. It is clear that educational goals alone do not drive the educational-software development process."⁹⁹

Some educators are fascinated with *any* use of computers, even applications in which the computer is incidental. In the report "Transforming American Education: Reducing the Risk to the Nation," the National Task Force on Educational Technology speaks glowingly of using computerized databases to present already tabulated statistics about which fields of employment are expected to grow in the future; they assert that this rote use of computer technology will "show students the impact that computers are having on the working world they will be entering."¹⁰⁰ One teacher created a database about famous people; for one assignment, students paged through the database, reading one file after another and copying information verbatim onto their homework forms.

⁹⁹Adeline Naiman, "A Hard Look at Educational Software," *Byte* (February 1987) 198.

¹⁰⁰Ridley and Hull, 9.

Another teacher talked about having "programmed writing programs"; these turned out to be word-processor files that the students paged through, exactly like a printed form, with blank spaces left for the students to answer questions. One school displayed a bulletin board of "Computer Work," which consisted of simple black-and-white drawings produced on inexpensive, low-resolution printers; in contrast, students' freehand drawings were complicated, varied color images. Responding to the special attention given to simulation software, some software developers market trivial simulations of events that are much easier and better experienced directly; e.g., one very boring but inexpensive program "simulates" growing a plant. In these applications of computers, the computer is an irrelevant and unnecessary component.

The pressures that drive school computerization serve to narrow the focus primarily to the technology itself. Much of the time, teachers and administrators are under such strong pressure to use computers, that they feel forced to select hardware and software before they have time to even consider what to do with computers once they get them. This is one reason curriculum development and educational planning for computer use lag far behind equipment purchases. *Schools concentrate first on buying hardware, then buying software, then training teachers, and only then – if they get this far and still have additional time and money – they consider curriculum.* In pursuing the steps in this order, they almost guarantee that any later efforts at curriculum planning will have to accommodate the technology, rather than requiring that the technology serve educational needs. But this sequence of steps makes perfect sense if one realizes that as soon as schools buy computers, any computers, they are relieved of much of the considerable pressure on them to be "part of the information technology."

Perhaps the most disturbing effect of the pressure-driven nature of educational computing is the restriction of what are considered appropriate questions to ask about computers and education. Despite the lack of evaluation, the assumption that computer education is worthwhile, even essential, is rarely challenged:

"People who have become accustomed to the 'literacy' idea find it very difficult even to entertain the question of whether there is any universally required computer experience. ... Merely to say the phrase 'computer literacy' defini-

tively answers a question which has not explicitly been asked. ... One practical result of the literacy metaphor is that many decisions about computer education have been made in a kind of panic."¹⁰¹

Indeed, some computer-literacy enthusiasts appear never to have questioned *whether* it is important to teach about computers in schools, or why. In the current climate of school computing, fundamental, difficult questions about the appropriateness of computers for furthering educational goals are rarely asked; instead, the focus is on superficial, relatively easy questions about what hardware and software to buy. A primary-school teacher explained it this way, when I asked why there is so much fuss about the need to have computers in schools: "I think part of it's trying to put a bandaid on what's wrong with the school system. Everybody realizes that they made a mistake, way back when. Somewhere along the line, they took a wrong turn."¹⁰²

It is unclear what percentage of educators genuinely believe that computer-literacy education is a worthwhile idea. Computer-literacy advocates are extremely vocal, and their enthusiasm dominates the education literature. But many teachers do *not* respond with pleasure to shiny new computers. These teachers' voices are rarely heard above the clamor in favor of computers in schools. One computer coordinator told me this: "It's like pulling teeth to get people to use computers."¹⁰³ There are signs that all the negative aspects of computers in schools – the unwanted pressures, the unfulfilled promises of potential benefits – are beginning to cause a backlash.

¹⁰¹Brian Harvey, "Stop Saying 'Computer Literacy'!" Article 9 in Terence R. Cannings and Stephen W. Brown, eds., **The Information Age Classroom: Using the Computer as a Tool** (Irvine, California: Franklin, Beedle & Associates, 1986) 45-46. Reprinted from *Classroom Computer News* (May/June 1983).

¹⁰²Interview.

¹⁰³Ibid.

Chapter 4

Computer Professionals' Perspectives

There is a noticeable absence of the computing community in evaluating and participating in computer-literacy education. Commercial development of new machines and software is influenced by strong market pressures that emphasize economic profit more than educational value. Research in "computers and education" is influenced by standards for success and promotion in computer science, which emphasize technical innovation more than educational reality. These standards foster new concepts, laboratory prototypes, and impractical (but technically interesting) visions, not systems that are likely to fit in actual schools (particularly at levels below undergraduate). Like most professions, the computing profession is not motivated from within to consider questions that might threaten part of its enterprise, such as fundamental, critical questions about the value of computers in schools.

The school system itself has no money to fund work in educational computing. Such direct funding might help ensure a better match between what technology can provide and what teachers want. Moreover, relatively poor schools cannot compete effectively with the commercial market and the military for the skills of trained computer professionals. Commercial and military funding sources have their own priorities and goals, which differ from those of the educational system.

As a group, computer professionals, the most computer-literate people in our society, know next to nothing about what is taught as computer literacy. Standard academic

training in computer science rarely includes anything about educational computing. A small number of computer professionals are involved in volunteer committees that advise schools, but these committees usually have a limited mandate; the decision to buy computers or to teach computer literacy has usually been made by the time such committees are formed and is not part of such a committee's work. For the most part, educators have little opportunity to learn what the computer profession thinks of computer-literacy education.

The few computer scientists who comment on computer-literacy education believe that the typical school-computing environment (where environment refers to a combination of hardware, software, and people) is too poor to support any worthwhile education. I have spoken with and surveyed a number of computer professionals about computer literacy.¹ Their criticisms begin with the basics, the hardware used in typical computer-literacy classes and the software that runs on such hardware:

"One thing that I noticed was that people tended to 'fixate' on whatever the computer they used at school was. ... The Apple-II is really quite a limited machine, and while it can be challenging to people attracted to computers the skills it teaches really don't help in the real world. ... People [are] fixating on yesterday's technology."

"Advances in electronics technology have made small personal computers inexpensive and widely available. But although their components are modern, their capacities and their software tools often reflect twenty-year-old technology. Formative exposure to these machines has some potential for harm. ... Of course, those who go on to bigger machines and bigger programs will soon learn about the problems [with computer software] I've been discussing, as will those who are educated in other ways. The rest, however, stand to be

¹Unless otherwise cited in footnotes, quotations in this chapter come from the electronic survey I conducted, described in Chapter 1. I contacted dozens of computer professionals, from recent graduates of computer-science degree programs to people who had worked in computing for decades. My references to "computer professionals" here include a range of people, including researchers, university professors, hardware and software engineers, programmers, and students.

misled profoundly about modern computer technology.”²

In addition to citing hardware that is unreasonably simple, small, and outdated, they often cite software that is simplistic and hard to use: “the hard way to do everything.” Yet few computer professionals know enough about primary- or secondary-school education to understand that the limitations of school budgets and priorities preclude school purchases of more powerful, more expensive hardware and software. While highly critical of current computer-literacy education, the computer professionals I spoke with had no suggestions of how to pay for better, widespread computer education.

Among the people I contacted, special criticism was reserved for programming instruction in computer-literacy classes:

“This year perhaps a million people will write their first program – probably a BASIC program for a small computer from Apple, IBM, Radio Shack, or Commodore – without anyone explaining to them that there’s more to program correctness than intuitive fiddling. They will think of themselves as joining the mainstream of the computer age, when in fact they are being introduced to programming as it was understood twenty-five years ago. ... To give people their formative programming experiences with a language [BASIC] that lacks the single most important concept of modern programming – abstraction – is ridiculous.”³

“The basic concepts are missing. No data structures, no input/output, no tool crafting, no *design*. Nothing to aid the student in solving a problem on the machine! It is sad.”

People I spoke with commented on the absence in computer-literacy classes of instruction in essential aspects of contemporary programming, including software-engineering concepts such as structured programming and information hiding, notions such as programming style and program correctness, the value of program documentation, and the

²John Shore, *The Sachertorte Algorithm and Other Antidotes to Computer Anxiety* (New York: Penguin Books, 1985) 183.

³*Ibid.*, 206.

necessity for programs to be written so as to be comprehensible to and maintainable by other people in the future. They decried the emphasis, in computer-literacy courses that include programming, on what they viewed as the unimportant mechanics of writing program code, instead of the much more important (and difficult) work of *designing* good programs. They wished that computer-literacy students who study programming were exposed to much larger programs, "so they can see what the real programming world does." Like many educators, some computer professionals believed that programming has no place in computer-literacy courses: "There is a lingering feeling that 'literacy equals programming,' which, for most people, simply isn't true. Literacy is in knowing how to *use* a computer. *Programming* is a vocational skill."

The programming component of computer-literacy courses received special attention from the computer professionals I spoke with, probably because most of them focused on the technical aspects of computer literacy, not the educational aspects, and programming is the most technical part of a typical computer-literacy course. Because it is the most technical part of the class, programming is also the part of the class that most highlights inadequate teacher training:

"From a professional point of view I have not been too happy with the programming teachers. They are usually not really trained in programming and tend to encourage poor practices."

"Teacher training must include the rudiments of software engineering, which appears to be the exception rather than the rule. If we give our children twenty-year old programming languages without giving their teachers the benefit of the past twenty years of computer science, we doom them to fall into the same bad habits we did."⁴

A number of other people also focused on the preparation of computer-literacy teachers:

"The problems I saw ... were that the instructors were essentially learning the subject haphazardly as they were teaching it."

⁴John Shore, "Computers at Capital Hill Day School" (Washington, D.C.: 23 September 1982) 18.

"Most of the [computer-literacy courses] I have seen are taught by a teacher that [sic] has had a 1-3 week computer camp for teachers, and knows little else."

"Secondary schools, when they have computer education programs, often use instructors whose own computer competence is questionable."

Computer professionals who had once been students in computer-literacy classes often criticized their teachers' as having been "one step ahead of the class at the time." Yet few computer professionals have concrete suggestions of how to improve the preparation of primary- and secondary-school computer teachers (let alone how to pay for better teacher preparation). In some cases, the people I spoke with felt that more detailed technical information should be taught in computer-literacy courses, which would require teachers to have even more training than is required (and not provided) now.

Generally, computer professionals tend to view current computer-literacy education as a superficial treatment of a vaguely defined topic:

"My impression of computer literacy in general is that it is vastly oversold and underdefined."

"I find 'computer literacy' to be a buzzword in search of a definition."

"The high school 'computer literacy' programs that I have seen have been a joke."

"Computer literacy courses are garbage. ... Computing is given a cheap, cursory, trendy treatment I suspect that [it] actually does more damage than good."

Daniel McCracken, a leading computer scientist, calls computer-literacy education "kindergarten-level training," soon to be obsolete:

"I also look to historical precedents, where I see no other attempt to educate the entire populace about new technology having (so far) much greater impact on people's everyday lives than computers, such as automobiles and TV. The

only direct precedent was the New Math fad a couple of decades ago, which was pushed for many of the same reasons now being advanced for computer literacy, and which sank with hardly a trace when those arguments turned out to be mostly empty. ... I don't understand what defense can be made for having all students ... take a required course in which they write a tic-tac-toe program and a program to find all the primes less than 100, and in which they are spoon-fed some pabulum about societal impact."⁵

Alan Kay, a seminal researcher on easy-to-use personal computers, believes that the typical school approach of learning about computers by emphasizing how to get things to run is like learning language by emphasizing how to recognize words and books. The flaw in such education is too much emphasis on mechanics and not enough on higher-level content.⁶ Some educators agree: The principal of a K-12 school in North Carolina believes that only computer applications, not mechanics, should be taught, and only on a "need to know" basis.

Perhaps the most scathing indictments of computer-literacy instruction come from college computer-science teachers, some of whose students have taken computer-literacy courses before entering college. I spoke with a Harvard professor of introductory (freshman) computer science, whose students include many graduates of the best primary and secondary school systems in the country, most of whom had prior exposure to computers. He had this to say about their preparation: "Secondary school education is doing nothing for these people. They don't know anything about computers. It's a waste of time." He added that these students have no advantage over other students who never used a computer before.⁷ Other university professors I spoke with echoed his comments:

"My experience is that students come to the university woefully prepared for any kind of encounter with a computer."

"There's no question that the college students I see ... are familiar with

⁵McCracken, 1, 4.

⁶Alan Kay, "Computer Software," *Scientific American*, 251, 3 (September 1984) 59.

⁷Interview.

computers ... in about the way they are with toasters. Few have any idea of what they can do with the machine."

"The worst problem, by far, for most students was their secondary-school training. Questions focusing on problem decomposition were met with queries like 'Ah, so I should use a FOR loop, right?' They had been trained to believe that programming was the job of translating a solution in pseudo-code into BASIC. The idea that they should participate in the solution was foreign to them. ... It took about 3 weeks for the students to overcome their inadequate model of programming. ... The key problem was that someone in the schools had decided that computer programming should be taught (probably to increase computer literacy). So, the history teacher or math teacher who had once been to a Radio Shack was told to get the best deal on some Apple-IIs and go with it. The teacher picked up the manuals and a BASIC book from a local bookstore and read them to the class."

Despite all this strongly worded, clear-cut criticism, computer professionals give conflicting messages about the underlying value of computer-literacy education. They complain about almost all aspects of current computer-literacy education and, when asked about educational priorities, most argue that computer literacy merits a low priority. But when asked about whether computer literacy should be taught, their responses are strongly positive, serving to bolster already strong pressures on educators to add computer-literacy instruction to school curricula: "Clearly, our responsibility to educate our children includes the responsibility to educate them about computers."⁸ The ACM (Association for Computing Machinery), the largest professional organization for computer professionals, formed a Task Group on Computer Literacy, which reported this in 1981:

"It [computer literacy] is an honorable goal with few negative consequences. It is seen as providing a basis for improving national productivity and the

⁸Shore, "Computers at Capital Hill Day School," 2.

well-being of all citizens. It is one of the cornerstones of a society built on technology. It is, according to some, necessary for survival."⁹

Consider the range of explanations given by computer professionals for why computer literacy should be taught:

"... there's no doubt in my mind that skill with a computer is vital for professional competence in today's world, and is becoming more so."

"The most important computer literacy skill is logic, and it just so happens that logic applies to much more than computers. ... If a child can leave elementary school, or perhaps to be more realistic junior high school, with a grasp of some of the fundamentals of reasoning, it would greatly improve their [sic] ability to think logically about the subjects their high-school teachers will throw at them – not to mention what the real world will throw at them."

"I think that computer literacy is a subset of something much more important in our society, namely critical thinking"

"I do believe that studying computers and programming has helped me to analyze and solve problems. And I believe that the beneficial effect extends beyond computer problems. I suspect that the effect is due to the practice you get – when you practice disciplined thinking in the context of computer problems, it helps with disciplined thinking in the context of non-computer problems. ... Simply put, mental exercise is mental exercise."¹⁰

"Given the exponential increase in the use of computers in every facet of life (both in our everyday and working lives), some degree of 'computer literacy' will become an essential component of the minimum requirements for success in even the most banal endeavors (such as, for instance, shopping)."

⁹Randhawa and Hunt, 2. Quoting D. Klassen, "Computer Literacy," in *Topics: Computer Education for Elementary and Secondary Schools* (New York: ACM, January 1981) 66.

¹⁰Shore, personal communication.

Many computer professionals believe that "computer literacy" should be taught. Their reasons for this fall into the same categories most often given by others for teaching computer literacy:

1. Computer job skills will be needed by everyone,
2. Learning about computers is good discipline for the mind.
3. Every informed citizen in today's society must know about computers.

These reasons are explored in detail in Chapter 5, where I argue that there is no compelling justification for them.

In thinking about computer-literacy education, some schools establish advisory committees that include local-area parents who work with computers and volunteer to help the school. Such committees also offer contradictory advice to schools. On the one hand, they emphasize the need for significant curriculum development, teacher preparation, resource management, and overall planning, all *before* any selection of hardware and software. One advisory-committee report argued that before introducing computers, a school should have a large-scale staff-development plan, and before choosing software, a school should develop curriculum goals, then choose software to accomplish those goals:

"One question must first be answered: What is the purpose of computers in our schools? ... It is only an idealistic hope that computers themselves will improve the educational experience for students in the schools."¹¹

The reports' authors noted that school plans for computer literacy need to be reviewed often (they recommended reviews every 6-12 months), because the technology changes so quickly. They pointed out that their town's schools had already bought computers "but have yet to undertake the serious planning and development that distinguish a comprehensive program from fragmented, piecemeal tinkering."¹²

¹¹Daniel R. Ries, "The Role of Computers in Belmont Public Schools: Final Report of the Advisory Committee to the School School Committee on the Use of Computers" (Belmont, Massachusetts: February 1984)) 3.

¹²*Ibid.*, 2.

But the committee was severely limited by the town in the sort of recommendations they could make. Before setting up the advisory committee, the town had already purchased computer hardware and software. The committee's mandate was limited to recommending how these resources should be used to teach computer literacy. The committee was not free to consider the priority of computer-literacy education or whether the school should teach it.

It is typical for policy-making about computer-literacy education to be split into two distinct types of decisions, handed over to two distinct sets of experts. Educators make decisions about whether computer literacy should be taught, and they consider questions of teacher training, curriculum development, and so on, to be their domain. They may reserve for advisory committees and computer professionals the task of making recommendations about technical details, especially about what particular hardware and software to purchase. Some committees may still point out the importance of issues other than technical ones, but they usually end up supporting the school's already-made decisions by concluding that computer-literacy education is a good thing. One report argued that "it would be unwise and perhaps counterproductive in the long run" to defer resolution of issues such as curriculum development, training, and long-range educational planning, but it stated that "installing computers without resolving these issues would be better than doing nothing at all"¹³ Given the tremendous pressures on schools to install computers, such statements are sufficient to serve as certification from the computer community of the value of computer-literacy education.

Where there is strong criticism from computer professionals about the basic idea of computer-literacy education, it is found not in school advisory-committee reports but in the computing literature:

"On our present course, we are going to waste ten years, a lot of money, and an irretrievable opportunity to do more useful things - on a venture with goals that I do not believe even its proponents have clearly defined. The attitude seems to be, 'The Computers Are Coming! Don't just stand

¹³Shore, "Computers at Capital Hill Day School," 28-29.

there! DO SOMETHING!' Well, sure, the computers *are* coming. They will affect our lives, and the lives of our children and our grandchildren, in ways more profound than any previous technological innovation. But, much as it hurts me to say this considering that I write programming books for a living, I don't understand how we prepare for that future effectively by requiring all students to undergo kindergarten-level training in the way computers are currently programmed."¹⁴

But the computing literature is unlikely to be read by many computer-literacy teachers.

Finally, I talked with computer professionals about alternative instruction in "computer literacy." A few people proposed more detailed technical education (one suggested "elementary digital technology the idea of building complicated things out of very simple logic circuits might be something that everybody should know"). Most of the suggestions focused not on technical detail, but on the social implications of computer use:

- *Limits of computer software*; e.g., computers can do only what people understand well enough to describe in a computer program.
- *Attack the myth of computer infallibility*. "I would hope that they would start teaching that the reliability of the computer is only as good as the software. A computer is not infallible, even though the hardware may not make mistakes. If the programmer doesn't think things through, the shortcomings s/he creates may live forever. And don't just accept something as gospel because it came from a computer."
- *Risks of trusting computers inappropriately*. One computer scientist felt that "everyone should have certain 'true' perceptions about computers," which must include a large dose of skepticism and the understanding that it is inappropriate to trust computers in certain situations, notably, situations that people understand poorly.

¹⁴McCracken, 5.

- Ethics. A school advisory-committee report noted: "The ubiquitous computer requires that educators identify and discuss the philosophical and ethical issues that this technology raises."¹⁵ A computer engineer said this: "There is a great lack of emphasis on the need for ethical behavior in the the use of computers. Just as we like to tell children of the dangers of alcohol and drugs in school, we need to tell them of the dangers of unethical use of computers."

Several people urged that "computer-literacy" courses include examples of positive uses of computers, not just warnings about computer uses. In fact, there is already an abundance of positive presentations of computers, in classes, advertisements, and the media. What is in short supply is the intellectual balance that would be provided by informed criticism:

"The public ... regularly receives highly biased and misleading information on computers and computer reliability. The nature of our commercial and research society is such that we only talk about our successes, never our failures and limitations. When a new program works, we broadcast that information. When a program fails, we stay silent. When a project succeeds, we present it at conferences and tell the newspapers. When a project fails, we go on to new things, wiser but with little fanfare."¹⁶

¹⁵Ries, 9.

¹⁶David Parnas, letter, 29 July 1985.

Chapter 5

Why Is Computer Literacy Taught?

A middle-school teacher responsible for her school's computer-literacy classes says she is not an optimist about computers in schools, but she believes it is important for schools to teach about computers. When I ask why she believes this, she gives three reasons: job skills; "it is another example of a system, a hierarchical way of thinking"; and "computers are all around, so everyone has to interact with them."¹ Her reasons are the most common ones offered for teaching computer literacy in schools. I describe them as follows:

1. The argument of jobs. Computer job skills will be needed by everyone,
2. The argument of mental discipline. Learning about computers is good discipline for the mind.
3. The argument of informed citizens. Every informed citizen in today's society needs to know about computers.

Claims that computer-literacy education is the ticket to a good job, "better" thinking, and wise public policy are widely accepted but rarely justified. Nowhere in the literature about computer literacy does anyone offer – or even demand! – justification for them. The reasons given for teaching computer literacy do not stand up to close examination.

¹Interview.

5.1 The Argument of Jobs

In 1984, Senator Chris Dodd of Connecticut said that learning about computers could improve a person's job prospects, because "by 1990 an estimated 30 million jobs in a broad range of fields will be computer related."² John Kemeny put it more dramatically in an interview:

Kemeny: "I believed [in 1963] that knowing how to use the computer was as important as reading and writing."

Question: "Do you still believe that's true?"

Kemeny: "Absolutely. In fact, I'll take it a step further. In the future, people who are computer *illiterate* will be as unemployable as people who don't know how to read or write today."³

Educational materials for and about computer-literacy courses assert that in the future, people without computer skills will be at a competitive disadvantage in the job market:

"... an extensive study in the United States concluded that for the year 1967, 46 percent of the gross national product was produced by information industries and that nearly half of the labor force held information-related jobs It goes without saying that the future work force will be expected to possess computer literacy"⁴

In Belmont, Massachusetts, a committee set up to advise the school on its use of computers (not *whether* computers should be used, but *how*), reported this:

"... half the work force is dependent upon information systems to carry out their job task. ... Indeed, computers have become commonplace, from the executive suite to the loading dock and from the supermarket check-

²Joseph A. Menosky, "Computer Worship," *Science* '84 (May 1984) 43.

³Roberts, *Scholastic Computing*, 41.

⁴Randhawa and Hunt, 6-7. Citing M.U. Porat, *The Information Economy*, Washington, D.C.: U.S. Government Printing Office, 1980.

out counter to the kitchen counter, elevating the importance of computer literacy in many professional and non-professional occupations.”⁵

One computer-literacy handbook states: “It will only be a short time before *all* jobs are computer dependent”⁶

It is a short leap from these strong statements about the omnipresence of computers in the workplace to the assertion that all students must be taught how to use computers. Computer-literacy education, it is argued, will give individuals the job skills necessary to be productive members of today’s technological workplace. And computer-literacy education will give the country the appropriately skilled workers necessary to regain our economic competitiveness.

The argument of jobs has several rebuttals:

1. Most jobs do not use computers, and very few jobs require significant knowledge about computers,
2. Jobs that “use” computers rarely require knowledge about computers, and
3. The skills that businesses really want are not related to computers.

First, most current and projected jobs are not dependent on computers. Labor-market studies conducted by the U.S. Bureau of Labor Statistics and others indicate that the occupational outlook is for more jobs in menial service areas. Some of these jobs may “use” computers, but they will not require any knowledge of computers; indeed, they will require little or no training beyond high school.⁷

Few jobs are genuinely “high-tech” in the sense of requiring significant knowledge of computers. While articles about computer literacy imply that jobs abound in high-technology industries (and that these jobs require computer-literate workers), research

⁵Ries, 19.

⁶Deborah L. Brecher, *The Women’s Computer Literacy Handbook* (New York: New American Library, 1985) 9.

⁷See U.S. Bureau of Labor Statistics, “Occupational Outlook Handbook, 1982-1983,” Superintendent of Documents, Washington, D.C.: U.S. Government Printing Office, Bulletin No. 2200. Gene I. Macroff, “The Real Job Boom Is Likely to be Low-Tech,” *New York Times* (4 September 1983) 16E. Langdon Winner, *op cit.*, 106.

suggests that the number of high-technology jobs is far fewer than most people think. Contrary to the popular belief of computer-literacy advocates, most jobs (even in high-technology industry) require very little in the way of computer skills or awareness.⁸ According to an economist with the Communications Workers of America: "Most jobs regarded as high technology positions are characterized by routine work Workers must have only a high school education and ability to pass minimum intelligence tests."⁹ Jobs such as air-traffic controller are dependent on computers but not on knowledge about operating computers.

Judging by employment studies, computer companies will be hiring many fewer programmers than janitors and clerical workers. Levin and Rumberger studied employment and employment-growth projections for the 1980s. They concluded:

"Of the 20 occupations expected to generate the most jobs in the economy during this period [1978-1990], not one is related to high technology. ... Only 3 or 4 of the 'top 20' occupations in terms of total contribution to job growth require education beyond the secondary level In fact, more new jobs for janitors will be created than new jobs in all the five occupations with the highest relative growth rates. ... Revised BLS [Bureau of Labor Statistics] estimates show that high technology occupations, as a group, will account for only 7 percent of all new jobs between 1980 and 1990."¹⁰

A study by Carey reported that of the fastest-growing occupations, two were related to computers: data processing machine mechanics and computer system operators, both of which are relatively low on the status scale for computer-related positions. Taken together, these two positions account for only 1.3% of all occupations.¹¹ Jobs such as these do use computers but require relatively little training or training on specialized

⁸Peter Kelman, "Computer Literacy: A Critical Re-Examination," *Computers in the Schools*, 1, 2 (Summer 1984) 5.

⁹W. Serrin, "Computers divide AT&T and its workers," *New York Times* (18 November 1983) 1.

¹⁰Henry M. Levin and Russell W. Rumberger, Russell W., "The Educational Implications of High Technology," Project Report No. 83-A4, Institute for Research on Educational Finance and Governance, School of Education, Stanford University (February 1983) 4-5.

¹¹Max L. Carey, "Occupational Employment Growth Through 1990," *Monthly Labor Review*, 104 (August 1981) Table 2.

systems that is best learned on the job.

Genuine high-technology jobs that demand significant computer skills *are* increasing at a fast rate, but only a fast *relative* rate; that is, relative to the current number of such jobs, not to the job market as a whole. Carey also projected that the fastest *absolute* growth would occur in occupations unrelated to computers: janitors, nurses' aides/orderlies, sales clerks, cashiers, and waiters/waitresses.¹²

A second rebuttal to the argument of jobs is this: Jobs that "use" computers do not automatically require computer literacy. For instance, a secondary-school teacher raised the argument of jobs in an interview:

Question: "You've said that computers cost a lot of money, and you weren't impressed with the software that you got to see What makes you say computers are important?"

Answer: "Well, I think you'd have to be a moron not to see the handwriting on the wall. Computers are coming in. To what degree is hard to tell yet. Even for these kids that don't want to go to college, they want to get a job somewhere ... you can't go to the grocery store or anywhere else, the cash registers are computerized."

Question: "How much do you need to know about computers to work at the grocery store?"

Answer: "Oh, nothing."

Senator Dodd's figure of 30 million computer-related jobs includes grocery-store checkers using bar-code readers and other jobs that strain reasonable definitions of being "related" to computer technology. These people comprise the "overwhelming majority of the 30 million," and they need no computer-literacy instruction to do their jobs.¹³

One computer-literacy textbook describes how truck drivers "use" computers: They wear small computerized devices on bands around their heads, which beep if they fall asleep while driving.¹⁴ These truck drivers do not need to know how to use computers!

¹²Ibid., Table 2.

¹³Menosky, 43.

¹⁴Roberts, *Scholastic Computing*, 243.

Neither do car mechanics using computerized diagnostic systems, telephone operators using computerized directories, airline agents using computerized reservation systems, and cashiers using computerized cash registers. (The cashiers do not even need to know how to read, if they work for a company that uses the latest in graphic-icon-based user interfaces.)

High technology *will* affect many jobs, and many people will “use” computers, but few people need computer skills. Secretaries will use word processors, bookkeepers will use spreadsheets, and clerical workers will use record-keeping systems. But use of these new tools will not require workers with more sophisticated computer skills. The trend in computing is toward machines that are easier to use, that require less and less “literacy” of users.

A premise underlying the argument of jobs is that increased automation demands increased job skills levels. This premise is disputed. James Bright, a Harvard Business School professor, studied the effects of automation on job-skill requirements in a variety of U.S. firms, examining claims made over the past 20 years, that increasing levels of automation would require increasing skills by operators and other workers. Bright found that skill requirements first increased, then decreased sharply as automation increased. Aggregate skill levels of U.S. jobs have not changed much in the last 20 years, despite widespread automation:

“Indeed, one of the major purposes and effects of high technology is to simplify or reduce the skill requirements for performing a particular work task. With the exception of a relatively small number of highly specialized positions for designing and implementing high technology applications, most jobs will not require higher skill levels. ... The general educational requirements for creating good citizens and productive workers are not likely to be altered significantly by high technology.”¹⁵

¹⁵Levin and Rumberger, 11.

Today's automobile is much more sophisticated technologically than yesterday's but also easier to use. The same can be said about the many devices we use that contain computers.

Finally, the argument of jobs is used to justify computer-literacy education indirectly, as a means to provide students with other, "higher level" skills that, it is argued, businesses desperately need in their employees. Marc Tucker, chairman of the NAEP's Committee on Computer Competence, believes that "A generation of kids coming out of schools who were really skilled with data and had a grasp of complex dynamic systems would stand a much better chance of securing the economic future for themselves and their country than the generation that is now emerging from our schools."¹⁶ But his premise – that computer-literacy education gives students a "grasp of complex dynamic systems" – is unjustified. Another computer-literacy enthusiast says that "Accurate and accessible information is becoming increasingly central to economic prosperity, and the most productive employees will be those trained to refine knowledge rapidly from data."¹⁷ But his premise – that computer-literacy education teaches students how to better refine knowledge from data – is unjustified. (See the next section for a detailed discussion of the relationship between computer skills and thinking skills.)

Some computer-literacy advocates believe that because computers are "information-processing" machines, everyone who wants to use information must use computers: "... knowledge of and ability to use a computer will become a necessity for most citizens in the future. ... In the age of information, the ability to use computers will become a major tool of people who acquire and use information. That means most of us will use them since we will all acquire and use information."¹⁸ This argument confuses two quite different meanings of "using information." One way of expressing the difference is to say that computers process information syntactically, and people can process information semantically. Computer-literacy classes expose people to tools for calculating tables

¹⁶"Study Finds Children Know Computer's Parts, Not Whole," *The New York Times* (7 April 1988) A:24.

¹⁷Christopher Dede, "New Technology and Education," *Encyclopedia of Educational Communications Media Technology*, 2nd edition (Winter 1984) 3. Pre-publication copy.

¹⁸Jerry Willis, "Educational Computing: Current Status and Future Directions," *Computers in the Schools*, 1, 1 (Spring 1984) 6, 9.

of formulas, manipulating words and paragraphs, and searching and sorting lists. But people must decide what information is important and how to organize it to get a job done. It is people who are expected to make decisions based on information, some of which may be processed by a computer. Decision-making is a complex process that resists complete automation, despite the best efforts of computer scientists. What these efforts have taught us is that decision-making depends as much on human experience and unanticipated events as it does on algorithmic processes.

Surveys of businesses indicate that the skills they *do* want, but rarely find, in employees – written and oral communications skills and interpersonal skills – are unrelated to computers. A study by the Bell System showed a preference for managers who were good at organizing, planning, and decision making, abilities that the study correlated positively with liberal-arts educations.¹⁹ Business executives (in another poll) cited the following as the most important job skills they hoped to find in those they hire: communication skills, analytical ability, interpersonal skills, mathematical skills, and behavioral traits and attitudes related to sound business practices.²⁰ Levin and Rumberger reported that “Both U.S. and British employers indicate that they seek new employees with a sound education and good work habits rather than narrow vocational skills.”²¹

“Despite the assertions by some computer literacy advocates, even in today’s technological society, most business people are more concerned that their employees come to them with strong basic communications skills and good work habits than with a high level of specific technical skill. ... While businesses do not wish to spend money and valuable time teaching their workers basic communications skills or inculcating proper attitudes, they are thoroughly prepared to provide specific training to handle the machines, equipment, instruments, etc., required for the job. ...

¹⁹Roger Robin Ekins, “Computer Illiterate, and Proud of It,” *AACJC Journal* (October/November 1985) 30-31.

²⁰*Ibid.*

²¹Levin and Rumberger, 12.

Historically, job-specific vocational training in public schools has been a colossal failure. ... Schools do not need yet another area in which they are charged with failing the society. It would be better to acknowledge from the outset that computer literacy programs in schools have only limited value as prevocational training.”²²

According to one manager who is himself a computer professional, “computer-literacy graduates are useless.” Most of all, he told me, he needs employees who can *adapt* to changing technologies.

Business managers do talk about the need for more highly skilled workers. But the skills they are referring to are reading, writing, basic math, social interaction, and good work habits (such as showing up to work on time). *These* are the skills needed to succeed in today’s high-technology society, and they are lacking in many school graduates.

The perceived job-market need for computer literacy is largely artificial. The insistence on widespread, mandatory computer-literacy education *generates* a need, in the form of requirements for high-school graduation, college application, teacher certification, and job application.

“Just as good spelling is now required to fill out an application for a janitor’s job and a high school diploma is needed to become a nurse’s aide, and a college diploma is necessary for just about everything else, so tomorrow CL [computer literacy] will be a requisite for many jobs that actually require no computer knowledge. With such credential barriers in place, CL will have created its own necessity, and employability will depend in part upon a parcel of useless knowledge about computers.”²³

Though computer literacy is not a new basic skill, by saying so we make it so.

²²Kelman, 8-9. Citing Philip Shenon, “Industry asks basic skills,” *New York Times* (4/24/83) 12:49.

²³Noble, 53.

5.2 The Argument of Mental Discipline:

'Wheaties of the Mind'²⁴

Simply put, the "argument of mental discipline" is that "'electronic learners' will think on higher levels than their pencil-and-paper equivalents did previously."²⁵ The mental-discipline argument rests on the widespread association of computer technology with certain attributes that are also associated with thinking (in particular, with problem solving). These attributes include logic, rationality, organization, and procedurality, among others. It is widely believed that interacting with computers demands and nurtures "higher-level thinking skills" based on these attributes. For instance, several participants in a "Computers and Writing" course felt that their writing was better organized if they used a word processor, because they perceived word processors as highly organized programs. Invariably, the people who said this were the most well organized people in the class.²⁶

When I ask teachers what they are teaching in their computer-literacy classes, they often mention general skills such as problem-solving, logical thinking, organization, and planning. The Director of the University of Oregon's Center for Advanced Technology in Education (and a professor of Computer Science and Educational Psychology) has said that underlying computer-literacy education is the notion of an "effective procedure," which he defines as "a very systematic way of doing things."²⁷ One educator wrote that the purpose of teaching students to deal with computers is to cause "an increasing percentage of the members of our society to be able to do systems analysis, to specify step-by-step processes by which real-life problems can be handled in

²⁴Roy D. Pea, "Logo Programming and Problem Solving," Symposium on Chameleon in the Classroom: Developing Roles for Computers, 1983 AERA Annual Meeting (American Educational Research Association), Montreal, Canada, 26.

²⁵Carolee Matsumoto, "The Potential of Computers for Teaching Thinking," Article I in Terence R. Cannings and Stephen W. Brown, eds., *Telecommunications, Robots, and Problem Solving* (Irvine, California: Franklin, Beedle & Associates, 1987) 37.

²⁶Rosenberg, "Computers and Writing," 19.

²⁷Hawley, 3. Quoting Dexter Fletcher.

computers.”²⁸ A school superintendent of a California district said this about computer-literacy instruction:

“If the computer can help develop in the minds of these young people a whole logical form of thinking and problem solving, I think we’re going to bridge an extreme gap in all parts of the curriculum. ... If we can train the cells of the mind to think in an organized pattern, then all our teachings will be grasped at a much faster rate.”²⁹

A claim to “train the cells of the mind to think” is obviously extraordinary, especially when tied to a machine in such relatively short supply. When he made this statement, the superintendent’s district had an average of one computer for every 24 students, and students received an average of 30-60 minutes per week of computer instruction.

Some advocates of the mental-discipline argument cite vaguely defined, even nonsensical “thinking skills.” In 1985, a System Specialist in Boston’s Educational Computing Operations department said that computers in schools are worthwhile because they help teach “cognitive thinking development.”³⁰

A computer-literacy activity book describes two activities to teach the “skill” of “divergent thinking”:

Day 1: Make a list of tools and machines that extend human physical capabilities. ...

Day 2: Make a list of extensions to the capabilities of the human brain.

The book notes that students “will probably list aids such as calculators, books, robots, computers, watches, and so on.” The only clue to how the book’s authors define divergent thinking is this statement: “The more divergent the thinking becomes, the better.”³¹

Catalogs of “educational” software include categories such as reasoning skills, logical reasons, analytical reasons, and problem solving (along with, for instance, mathematics,

²⁸Anton Braun Quist, “‘Computer Literacy’: Wave of the Future – Or Relic of the Past?” *Computers in the Schools*, 1, 1 (Spring 1984) 68.

²⁹Menosky, 44. Quoting Ronald Flora, Superintendent of the Lake Elsinore, CA school district.

³⁰Pearlman.

³¹International Council for Computers in Education, 12.

language skills, computer literacy, and keyboarding).³² Commenting on such software, a 1985 Computer Coordinators' Committee on Computer Literacy said this: "The newest generation of problem-solving software offers a way of having students learn and practice gathering appropriate information, deciding what to do with it, and interpreting the results of whatever is done. The intellectual gains from problem solving in this way are certainly transferable to other areas outside the realm of computers."³³ As this committee's statement illustrates, advocates of the mental-discipline argument believe that exposure to computers will cause students to become "better" thinkers (where better is defined as more logical), *in domains other than computing*. In other words, the mental-discipline argument rests on the assumption that intellectual skills gained through the use of computers will generalize or transfer to other areas of learning:

"Most jobs will require workers with quite sophisticated higher order cognitive skills. For example, typewriters are now being replaced by word processing / information management / data networking devices, since one professional using such a machine can do the work of five clerk/ typists. ... However, this new device is not being used as was the typewriter, simply to input symbols. Instead, to realize productivity gains from the 'intelligent' tool, the employee must be well versed in complex pattern recognition, information evaluation, information synthesis, decisionmaking given incomplete information, flexibility, and creativity. These higher order cognitive skills are illustrative of an emerging altered definition of higher intelligence. ... Only a technology based educational approach can offer students direct experience with intelligent tools to build these new cognitive and affective skills necessary for work and family life."³⁴

Often, the logical-thinking benefit that allegedly derives from computer-literacy education is linked to a specific aspect of computer use, such as word processing. As

³²For instance, see "QUEUE: The Best in Educational Software" (Bridgeport, CT: QUEUE, Inc., 1987). See also "Sunburst Educational Computing Courseware" (Pleasantville, NY: Sunburst Communications, 1987).

³³"Mission: Define Computer Literacy. The Illinois-Wisconsin ISACS Computer Coordinators' Committee on Computer Literacy Report (May 1985)," 12.

³⁴Dede, 15-17.

mentioned before, some people believe that word-processing programs improve writing skills by training (or at least making it easier for) people to organize their thoughts "better"; however, the evidence is at best mixed. Researchers at the University of Minnesota found that students who adapted best to word-processing programs were those who were already accustomed to planning before writing – those people who began writing only after they had a good idea what they wanted to say and how they wanted to organize it. In contrast, students who did relatively little planning before writing and then needed to revise extensively did not perceive word-processing programs as particularly useful.³⁵ Commenting on this research, Colette Daiute, author of *Writing & Computers*, wrote this:

"Despite the ease of revising and editing, the subjects who reported that they did not like to revise still didn't revise. This research has ... reminded us that the writer – not the computer – does the difficult conceptual work. Contrary to reports by many professional writers, the students who revised extensively did not find much solace in the easy block move and automatic re-copy commands. This may be because the precursor to using these commands – thinking about how the text should be pruned, expanded, and reorganized – is hard work."³⁶

Daiute's own research indicates that writers who revise their text with word processors tend to rewrite and reformulate *less* extensively than writers who revise with "older technologies" (pens and typewriters). Writers in her study made only small changes with word processors, but they made "substantial improvements" when they had to spend more time recopying their text.³⁷

Word-processing programs may be valuable and convenient tools for good writers,

³⁵See L.S. Bridwell, G. Sirc, and R. Brooke, "Revising and computers: Case studies of student writers," in S. Freedman, ed., *The Acquisition of written language: Revision and response* (New York: Ablex, 1985). See also L.S. Bridwell, P. Johnson, and S. Brehe, "Computers and composing: Case studies of experienced writers," in A. Matsuhashi, ed., *Writing in real time: Modelling production processes* (New York: Longman, 1983).

³⁶Colette Daiute, *Writing & Computers* (Reading, Massachusetts: Addison-Wesley Publishing Company, 1985) 221-222.

³⁷*Ibid.*, 39.

but they do not address the most difficult tasks involved in writing. Like other computer applications presented in computer-literacy classes, word-processing programs deal only with the mechanics of a task, not its intellectual content. They help the writer manipulate characters, words, and blocks of text, but they do not give people good ideas or worthwhile things to write.

Most of the time the mental-discipline argument for computer literacy is used, it is linked to the teaching of computer programming. A prevalent assertion is that computer programming improves general thinking abilities:

"There is an undocumented but much discussed advantage that many believe to result from using computers and writing computer programs. The most popular term for this advantage seems to be 'procedural thinking,' and it refers loosely to supposed improvements in logical thinking and problem solving. I don't like the term - a better one would be 'disciplined thinking.' But although this effect is harder to achieve than is commonly supposed, I believe that it's real. The study of computers and computer programming can lead to improvements in your ability to analyze and solve problems. ..."³⁸

Computer-literacy coordinators who were surveyed said this: "Only a few students will go on to become computer programmers, but all students have something to gain by learning to write simple computer programs. Computer programming provides a logical, systematic framework for problem solving."³⁹ One educator wrote that computer-literacy courses should include instruction in programming, because "...the logic and problem solving skills learned are invaluable."⁴⁰ Another educator wrote: "Programming can serve this purpose [of increasing problem-solving and logic skills] well. It encourages, even demands, logical reasoning. It sharpens problem-solving abilities"⁴¹ Even some people who are highly critical of computer-literacy education accept without question that learning to program is a good thing for all students: "We should know how to do a little

³⁸John Shore, *The Sachertorte Algorithm and Other Antidotes to Computer Anxiety* 32-33.

³⁹"Mission: Define Computer Literacy. The Illinois-Wisconsin ISACS Computer Coordinators' Committee on Computer Literacy Report (May 1985)," 12.

⁴⁰Fritz, 707.

⁴¹Neibauer, 90.

programming and should understand something about the theories of computing. We should certainly engage ourselves in the marvelous problem-solving potential of computer technology and can only benefit from learning how to represent abstract ideas in concrete form."⁴²

Where computers are in short supply, the emphasis in computer-literacy education is on something related to programming, called "algorithmic thinking." One computer-literacy workbook, intended for classes with no computers, describes an activity that teaches the skill of algorithmic thinking. In this activity, groups of students walk around the classroom, stopping at several "learning stations." Each station has some playing cards and a set of instructions. In a sample learning station's instructions, the students are asked to count the number of cards at the station. If the number is odd, the students are instructed to record the word ODD on their Recording Sheets; if the number is even, they are instructed to record EVEN. They are then instructed to shuffle the cards and move on to the next station.⁴³

Advocates of the mental-discipline argument range from educators to computer salespeople to computer scientists, but none have any convincing support for the argument. Psychologists point out that transfer of cognitive skills between any two domains is exceptionally difficult to achieve or demonstrate:

"Merely because the same skill or ability level applies to two tasks, one cannot predict transfer with confidence. While two tasks may both involve a precise cognitive style ..., they may engage these in crucially different ways Transfer of problem-solving strategies between dissimilar problems, or problems of different content, is notoriously difficult to achieve even for adults."⁴⁴

When the manager of behavioral research for Atari computer company was asked about the basis for Atari's promotion of the educational value of its products, he said: "I guess you could say that the extent to which programming teaches generally useful mental skills is still an area open to question and exploration."⁴⁵

⁴²Ekins, 31.

⁴³International Council for Computers in Education, 5, 18-22.

⁴⁴Roy D. Pea, "Logo Programming and Problem Solving," 29.

⁴⁵Menosky, 46. Quoting Robert Olton.

Evidence that is relevant to the mental-discipline argument comes from the few studies done on the cognitive effects of learning programming. These studies are inconclusive, yielding apparently contradictory results. Several of the most careful studies, done at the Bank Street College of Education, uncovered no evidence to support the mental-discipline argument.

In one project at the Bank Street College, researchers compared elementary-school students who never programmed with students who spent 30 hours programming over the course of a school year (a substantial amount of time by school standards). The study focused on the effects of programming instruction on planning skills. It concluded that there was no difference between the programmers and non-programmers in all the dimensions of planning that were considered:

"There is no currently available programming language or computer environment that can, in itself, without instructional guidance, help students to develop these advanced thinking strategies, or make them aware of the broad range of problem domains to which they might be applied. At the current time, learning these skills is a deeply social affair"⁴⁶

Another study considered the development of thinking skills in three groups of high-school students: those with no programming experience, one year of experience, and two years of experience. There were no significant differences among the groups in several measures of "far transfer," or transfer from programming to relatively dissimilar activities. Significant differences were found only on some subscores for the "near transfer" task, algorithm design and analysis: The programming students were more likely than the control group to apply simple programming structures learned in their programming course (a loop, a conditional test, and a counter) to other programming tasks. Also, more programming students than non-programming students could write algorithms with only a few flaws (although few programming students could develop successful algorithms).⁴⁷

⁴⁶Roy D. Pea and D. Midian Kurland, "Logo Programming and the Development of Planning Skills," Technical Report No. 16, Center for Children and Technology, Bank Street College of Education (March 1984) 22-23, 44-45.

⁴⁷Kurland et al, 1986, 38-39.

While it is gratifying to note that programming instruction appeared to teach students something about programming, the researchers noted that the study's results provide no support for the widespread claims of far transfer:

"Although programming students were somewhat better on a task closely analogous to a programming problem, they were not better at reasoning with variables, developing equations, procedural means-end reasoning, decentering, or planning on tasks which were not like programming in terms of surface characteristics. ... The algorithm task, to which programming students did apply some of their skills, ..., bore relatively obvious similarities to a programming task Given the transfer of the operational skills here and not to our other tasks, it is apparent that relatively context-specific rather than general operations were learned. ...

From our perspective, based on data from the present study and others, we do not believe that the current hope for *incidental* learning of generalizable thinking skills through programming is realistic."⁴⁸

The same study reached some unexpected and troubling conclusions about the learning that did take place among students studying programming:

"Some students relied on the understanding of a few good students and never bothered to learn the material themselves. ... Though several were concerned with understanding what to do, they did not seem to have techniques or rules for systematically analyzing buggy programs and for developing corrections. Students often seemed to feel helpless. ... Unlike many other domains where the opposite is true, in programming a student's performance, as reflected in completed programs, can clearly outstrip understanding by a considerable margin."⁴⁹

Other studies echo these negative conclusions about the effects of introductory programming instruction:

⁴⁸Ibid., 44, 48-49.

⁴⁹Ibid., 43-44.

"[Students'] understanding of the program is basically at the level of a single line. ... Novices usually have a tinker toy model of program construction. They presume that programs are assembled by piecing the language features together. ... They do not engage in the activities required for refining the natural language statement of the problem into a statement which can be decomposed and coded into a problem solution. As a result, when they are asked to solve problems which are more complex than simple translations of known language features, their solutions are often poorly organized, incorrect, or inefficient. Thus, the top down design and stepwise refinement which experts use to write programs is not taught nor required by most introductory programming courses at the junior high level."⁵⁰

Researchers have noted that many students rarely do any planning before programming; instead, they go to their computer terminals and write programs from scratch. They do not debug their programs much either, preferring the tedious task of completely rewriting a program, rather than the intellectually harder task of figuring out what is wrong with the first program.⁵¹ I observed this behavior repeatedly in computer-literacy classes.

Other studies of programmers have been condemned for having serious methodological flaws, including overly small numbers of subjects and lack of a control group. In one project that found evidence of transfer from Logo programming, the researchers had studied groups of two or three students who were specially guided by an adult tutor.⁵² Were the effects noted by the researchers the result of Logo instruction or of an extraordinarily high teacher-student ratio and specially trained teachers?

Another project uncovered both positive and negative effects of Logo programming on problem-solving abilities. Some students appeared to be "learning" logical fallacies. But the Logo children tended to ask more questions and to self-correct their errors. However, the students learning Logo had received extra interaction from teachers and peers and

⁵⁰Dalby, Tourniaire, and Linn.

⁵¹Roy D. Pea, "Logo Programming and Problem Solving," 30.

⁵²Gavriel Salomon and D.N. Perkins, "Transfer of Cognitive Skills From Programming: When and How?" *Journal of Educational Computing Research*, 3, 2 (1987) 161.

much prompting with "metacognitively-oriented questions."⁵³ The other students – using CAI systems (computer-aided instruction) – received no such attention.

Some researchers conclude that it is precisely the extra attention that programming students often receive (in studies of programming and cognition), which accounts for any perceived benefits of programming instruction:

"Originally, one might have thought of programming as a kind of cognitive playground; mere engagement in the activity of itself would exercise the mind as real playground exercise young bodies, without any need for instruction finely tuned to provoke such consequences. Unfortunately, the research argues against such a vision.

Instead, according to our interpretation of the conditions for transfer, certain inconvenient conditions must be met. ... The straightforward teaching of elementary programming is not a very good way to foster cognitive skills."⁵⁴

According to this research team, the only likely way to harvest general cognitive benefits from programming instruction involves "a high teacher-student ratio, Socratic interaction with the learners, great sensitivity on the part of the teacher for the ebb and flow of enthusiasm and understanding in the individual student, calculated provocation of abstraction and connection-making, and so on." In the long run, the researchers suggest that improved curricula and teacher-training programs "may make possible" programming instruction that has an impact on cognitive skills.⁵⁵ As these researchers point out, satisfying this long list of preconditions for the mental-discipline argument would seriously strain the resources of the educational system. Any teacher would agree that large investments in teachers and curricula "may make possible" all sorts of wonderful educational results.

Additional studies of the effects of programming instruction have speculated on the relative importance of the programming and the instruction. One report on Logo stu-

⁵³Douglas H. Clements, "Longitudinal Study of the Effects of Logo Programming on Cognitive Abilities and Achievements," *Journal of Educational Computing Research*, 3, 1 (1987) 91-92.

⁵⁴Salomon and Perkins, 163-164.

⁵⁵Ibid., 164.

dents in elementary and high schools stated that although Logo *allows* practices such as planning, debugging, and careful documentation, students do not pick up these practices without explicit and well planned instruction.⁵⁶ As discussed in Chapter 2, the toy-like programming instruction in computer-literacy classes is neither well planned nor explicit about general cognitive skills such as planning.

Several meta-studies of programming and cognition have concluded that the alleged connection between learning to program and learning to think is unjustified:

"The field present[s] indecisive and apparently contradictory results. ... Overall the results [have] to be considered inconclusive."⁵⁷

"In the process of learning to program a computer, it is assumed, students will also learn about their own thinking processes. ... Despite these claims, there have been very few relevant research studies and almost no convincing support of this connection."⁵⁸

"The intuitively plausible claims for the cognitive benefits of programming have broadened in scope and in public attention. Although, as yet, there is no evidence to support these claims, their presumed validity is affecting important decisions in public education We run the risk of having the naive 'technoromantic' ideas become entrenched in the school curriculum by affirmation rather than by empirical verification through a cyclic process of research and development."⁵⁹

The mental-discipline argument is a great hope, but it remains an unsubstantiated hope. What little we know about the psychology of programming is inconclusive. What we know is not commensurate with the strong claims that are made for the cognitive ben-

⁵⁶Roy D. Pea, "What Will It Take to Learn Thinking Skills Through Computer Programming?" Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana, 6.

⁵⁷Salomon and Perkins, 150.

⁵⁸Richard E. Mayer, Jennifer L. Dyck, and William Vilberg, "Learning to Program and Learning to Think: What's the Connection?" *Communications of the ACM*, 29, 7 (July 1986) 605.

⁵⁹Pea and Kurland (January 1984) 2.

efits of learning programming. Such claims are characteristic of arguments for computer-literacy education.

Some computer-literacy advocates fall back on a common-sense version of the mental-discipline argument: It is *obvious* that programming demands "good" thinking skills and equally obvious that these skills must rub off in other areas. This sort of argument is used to justify the teaching of a great many things in schools. But there are equally common-sense counterarguments; for instance, it is probably not true that computer programmers are "better thinkers" than any other group of people.

Contrary to popular assumption, programming does not *require* good organizational or planning skills, although it certainly benefits from these skills (as do many other activities). People who are normally disorganized are likely to write disorganized programs, which can nonetheless work "correctly." Like any other type of work, programming can be done carefully or sloppily, and sloppy programming will not turn the programmer into a well organized thinker.

Like programming, writing can be and is routinely done by people who are poorly organized and do little planning. Regardless of whether it is done with pencil and paper or with a word processor, bad writing does not make the writer a better thinker.

To summarize, there is no convincing support for the claim that learning about programming fosters mental discipline or that whatever is acquired in computer classes transfers to other areas of learning.

5.3 The Argument of Informed Citizens

The last frequently used argument for widespread computer-literacy education is the argument of informed citizens. It is also the most vaguely worded argument, often consisting of little more than this assertion: "Computers are everywhere, so people must need to know about them in order to function." To participate fully in a society that "uses" computers in so many common applications, it is argued, citizens must be computer literate, and it is the responsibility of schools to provide this training. Richard Hersh, Dean of the Graduate School of Education at the University of Oregon, articulated this

position dramatically:

"If schools do not set up master plans for meeting these needs soon, they will be responsible for turning out a generation of technopeasants. ... It is up to educators to see that the next generation become masters – not serfs – of modern technology."⁶⁰

Advocates of the informed-citizens argument say that in the imminent future, people who are not computer literate will be at a great disadvantage in all areas of life; indeed, that the ability to use computers will be "prerequisite to everyday life situations."⁶¹

Swept up in this rhetoric, it is easy for teachers to become convinced that it their duty to teach computer literacy. A 6th-grade teacher said this to me: "We cannot, as educators, ignore this. We have to make our students aware of this as a tool. They will encounter it. We're going to be crippling them by not exposing them to it."⁶² Another teacher spoke at length about the lack of demonstrated value in using computers in schools, but he continued to say that students should be taught to use computers. When I asked him why he thought it was necessary to use computers, he replied: "It's an important life skill, because computers are part of everyone's life."⁶³

Indeed, in a typical day, a person may "use" computers in a car, wristwatch, microwave oven, coffee machine, telephone, VCR, and so on. But although computers appear to be omnipresent, it does not follow that we all need to know how computers work. There are other technologies that people "use" more often than they use computers, but people need not understand how all these technologies work. Dan McCracken, a former president of the Association for Computing Machinery, notes that a strong informed-citizen type of argument could be made for something we might call automobile literacy:

"Cars are of tremendous importance in everyday life, the national economy, and all kinds of public policy issues. Most people will spend *far* more time

⁶⁰Kelman, 4. Quoting Hersh, "Are Americans schools turning out technopeasants?" *Instructor* (May 1983) 27.

⁶¹Hawley, preface written by Philip K. Piele, Executive Secretary, Oregon School Study Council.

⁶²Interview.

⁶³Interview.

and money on cars than computers, ever. A person starting driving today has a 50% chance of being seriously enough injured in a car accident during his/her lifetime to require hospitalization, and one chance in 50 of being killed. Cars created our suburbs, ..., and created much of today's global economics and politics That all adds up to a matter of great importance, by any measure I can think of. ... Just as we don't require all drivers to know what a differential is, there is no need to teach all students how to program."⁶⁴

We learn to operate cars – and televisions, and telephones, and so on – incidentally, to the extent that we need to make use of these devices. Similarly, people can easily learn to operate word processors or spreadsheet programs, if they need to use them. But learning how to operate an application program does not make one literate about computers in any meaningful sense.

A better analogy for computers as they are used in today's society is electric motors. Both are often embedded components in larger systems. In a typical day, one might use electric motors in an alarm clock, electric shaver, refrigerator, car, stereo system, and vacuum cleaner. Yet most people function quite well without knowing how electric motors work, and no one talks about the need for electric-motor literacy. Like electric motors, computers are becoming ever more invisible components of commonly used tools.

Some versions of the informed-citizens argument emphasize policy-making:

"In an information society such as ours, widespread understanding of computer technology and the consequences of computer use and misuse is necessary, even required, for public policy-making. ... For the individual, an understanding of computer technology and uses is important because it reduces bewilderment about computers and promotes a balanced view of the computer's role in society. Such understanding also enables people to use and influence the design of computer-based social services and to develop informed opinions regarding particular computer applications having political, economic and social implications."⁶⁵

⁶⁴McCracken, 3, 5.

⁶⁵Anderson and Klassen, 41.

In *The Sachertorte Algorithm*, a book about computers that is both nontechnical and meaningful, John Shore writes: "Learning more about computers can also help people to participate intelligently in the formulation of public policy."⁶⁶

Computer-literacy advocates worry that students who are not "exposed to" computers will be "left behind" and may "feel helpless in the face of computer power, while those who have used a computer for writing, programming, or graphics understand the limits as well as the power of the machine."⁶⁷ But the "exposure" to computers that characterizes computer-literacy classes can give people only the most superficial control over computers. Students who are exposed to word-processing software may be able to control a computer enough to format and print their text – and this may be an extremely convenient skill to have – but they have learned nothing about computers that could help them control computer uses in society.

There is a relationship between understanding and empowerment. Intelligent public policies related to computer use cannot be made by people who do not have some understanding of computers. But computer literacy, as that topic is currently taught, does not provide the kind of understanding of computers that could contribute to wise policies. Computer-literacy classes may be doing *harm*, by turning out people whose model of computation is extremely simplistic but who nonetheless believe themselves to be "computer literate." Douglas Noble refers to the "propaganda of Computer Literacy":

"It introduces people to computers, gives them some hands-on experience, and deludes them into thinking that all computers are friendly and easily controlled because their little micro is so. In this way, CL mystifies in the name of demystification. ... In fact, it is much more likely that a focus on minimal technical competence, as in CL, will lead to a sort of pseudocontrol, a false sense that one has power simply because one can make a computer do a little something. ... Even if technical understanding were important for democratic participation, the minimal technical information available in such courses is many orders of magnitude removed from any significant un-

⁶⁶Shore, *The Sachertorte Algorithm*, 32-33.

⁶⁷Daiute, 16.

derstanding which might serve to enhance public deliberation.”⁶⁸

A little of the wrong information masquerading as technological literacy could lead to dangerously unwise decisions about using computers. People whose education about computer systems is limited to tiny programs are not equipped to make informed decisions about the massive computer systems that run real applications. Such large systems are *qualitatively* different than what students encounter in computer-literacy classes. Hence the danger of trying to make people “computer literate” by exposing them to “toy” computers.

5.4 Other Reasons

In addition to the three major reasons for teaching computer literacy, discussed above, there are several “minor reasons.” The minor reasons often surface after a discussion reveals a lack of justification for the major reasons. The minor reasons have more to do with how the educational system normally works than with research on education.

Some people argue that pressure from parents who want their children to learn about computers is reason enough to teach computer literacy. It is then instructive to ask why some parents want their children to use computers. Parents are under pressure themselves: They read dozens of newspaper and magazine articles about the alleged need for everyone to know about computers (the informed-citizen argument), and they see dozens of advertisements that explicitly link good parenting with the purchase of computers for one’s children. Parents who have spent hundreds or thousands of dollars on a home computer have a vested interest in computers; they are anxious for their children’s schools to validate the wisdom of their home purchases by making similar purchases for school use.

Parents often ask me, as they do many computer professionals, what kind of computer they should buy for their children. When I ask why they want a computer, they have no answer but to wonder why I am asking such a silly question. When I ask what they

⁶⁸Noble, 58, 62.

want to do with a computer, they have no answer, because there are few good uses for home computers. (An exception may be word processing, a convenient tool that might be useful for some children who are old enough to be writing papers regularly. See the last chapter for more on this application.)

It appears reasonable that schools should be responsive to parents' and communities' preferences for their children's education. But the first response should be a thoughtful examination of justifications, priorities, needs, and resources, not a headlong rush into implementation of a new curriculum item.

Other people argue that teachers' enthusiasm about computers would be reason enough to use computers in schools for a variety of applications, including computer literacy. Again, it is revealing to probe the basis of this argument; in this case, to probe why some teachers respond positively when given a computer in their classrooms. Is this response a reflection of educational value of computers, or is it the natural result of being treated to a much heralded, state-of-the-art tool, in an otherwise poor working environment? I spoke with teachers who were enthusiastic about their school's new computers because they would no longer have to prepare lesson plans on old mimeograph machines; their schools could not afford copiers but had purchased computers. Every school I visited lacked basic office equipment, clerical and other support services, private office space, and other features of working environments that most computer professionals take for granted. The president of the National Education Association complains that today's society treats teachers like very tall children.⁶⁹ An article in an education journal claims that "the computer has become the new 'toy' for a select number of teachers trying to escape the monotony of regular education and satisfy the public relations goals of principals and administrators."⁷⁰

Computer-rich schools are treated not only to new equipment but much extra attention, from university researchers, television crews, etc. One afternoon in a Boston primary school, I watched 2 students using computers, while 10 researchers observed, videotaped, audio-taped, transcribed, and photographed those students. In an environ-

⁶⁹ *America's Kids: Why They Flunk*, ABC television special program, 3 October 1988.

⁷⁰ Neibauer, 88.

ment – the U.S. public school system – in which 5,200 teachers are physically attacked by students each month,⁷¹ without that raising a public outcry, is it surprising that some teachers and administrators respond with enthusiasm to positive attention from the outside world, when that attention comes in the form of new computers?

Still other people argue that computer-literacy should be taught because students like computers, hence will learn better if classes use computers. A System Specialist with the Boston Public School System justified Boston's large computer purchases by saying that "computers have had a positive motivational effect."⁷² Some parents cite their children's interest in computers as evidence that it is educationally worthwhile to expose children to computers.

As already noted, some teachers are excited about using computers. When teachers (or parents) are enthusiastic about teaching a given topic or using a given tool, they communicate this enthusiasm to their students, who are likely to respond positively to *whatever* is taught. Teachers' relationships with students are extremely important in determining how well students learn. A sixth-grade teacher who used word-processing programs in part of her English class said that her students performed better when her teaching style included personal "conferences" with individual students, "as much for physical closeness as educational information. Someone may want simply to be next to a warm body and talk, and they use their writing as the vehicle to get that closeness."⁷³ In interactions between parents and children using computers, the presence of a computer may be the least relevant component:

"What little kids probably love most of all about software isn't the software at all but the chance to sit on Dad's or Mom's lap. Once they're snugly in place on the parental lap, kids love whatever is going on. ... 'Scaffolding' is how child psychologists describe what the parents are doing by providing support to the learning process"⁷⁴

⁷¹Thomas L. Martin, Jr., "Intellectual Capital Formation and the Education System," National Academy of Engineering, *The Bridge*, 17, 3 (Fall 1987) 13.

⁷²Robert Pearlman, 1985.

⁷³Interview.

⁷⁴Snyder and Palmer, 62.

Learning is a nurturing process. As with teachers, some students' enthusiasm about computers may have little to do with computers themselves.

Students who appear enthusiastic about using computers are not necessarily learning better because of that enthusiasm:

"Students ... fail to associate their difficulties in achieving a solution with their lack of planning. Students believe they need more terminal time, not that they need to plan, to solve problems effectively. Students find the on-line experience very motivating and they prefer to be on-line interacting with the computer, even if they are not making progress in solving problems."⁷⁵

Kenneth Komolski, then head of the Educational Products Information Exchange, a non-profit organization affiliated with Columbia University Teacher's College and Consumer's Union, referred to "the fun factor," which masks serious questions about the real value of learning with computers: "Parents go by and say, 'Oh, the kids are so involved.' But is any learning really taking place? ... there is no evidence that students retain material better - or even learn better than if they'd been drilled conventionally."⁷⁶

Moreover, many teachers complain that after the initial interest in the new machine wears off, students quickly grow bored with "educational" software:

"But with most so-called educational software, says Gay Reetz, a Scarsdale, New York, grade school teacher and computer literacy instructor, that attention fades quickly. 'They do their multiplications or whatever, and the buzzer goes off and on, and some little man tips his hat, and they think it's great,' Reetz says. 'They learn the music, they listen for the buzzers, but once they have that down, the children are bored.'"⁷⁷

By 1986, Kim Natale, a high-school physical teacher who was the 1984 Colorado Teacher of the Year, had this to say about students' reactions to school computing: "Two or three years ago, the kids were all excited about using the computers. Now they say, 'Oh, do

⁷⁵Dalby, Tourniaire, and Linn, 11, 17.

⁷⁶Menosky, 45.

⁷⁷Ibid.

I have to?"⁷⁸ Students respond negatively to many of the same factors that teachers of computer-literacy courses complain about. For these and other students, enthusiasm about computers has already begun to wane.

Finally, some people agree with me that there is no good justification for teaching computer literacy, but they argue that computer literacy is no different – no *less* justified – than any other school topic. Some of these are the same people who argue that computer-literacy education is the necessary foundation for informed, critical decisions about computer uses. Yet, they themselves take a position about computer uses in schools that they cannot defend. If “computer literacy” means anything, surely this position is the epitome of computer illiteracy! Such an argument tells us much more about the problems of the country’s education system than about the desirability of computer-literacy education.

⁷⁸Levine, 78.

Chapter 6

Conclusions

The current optimism about computing is the most recent manifestation of an old fascination with technology. Langdon Winner, a political historian, notes: "It is not uncommon for the advent of a new technology to provide an occasion for flights of utopian fancy. During the last two centuries the factory system, railroads, telephone, electricity, automobile, airplane, radio, television, and nuclear power have all figured prominently in the belief that a new and glorious age was about to begin."¹ Society has not looked very critically at many proposed new uses of technology until later stages of technological development, when negative effects often are all too obvious. The computing profession is notorious for making overly optimistic claims about the capabilities of computers and the benefits of computer use. At the last Technological Literacy conference, Richard Devon pointed out: "Typically, these claims for the importance of computers rest on prophecy rather than on analysis of the present. However, in retrospect, predictions for computers have not proven accurate."²

The education system has repeatedly seen new technologies as its salvation. Computers are the latest in a long line of "educational technologies," following, for instance, radio, phonograph, stereoscopes, film, adding machines, reading machines, teaching machines, and television. Many of these now gather dust in school storage closets and appear on

¹Winner, 106.

²Richard F. Devon, "In Praise of Computer Illiteracy." *Bulletin of Science, Technology & Society*, 7, 1 & 2 (1987) 338. Presented at the Second National Conference on Technological Literacy, Washington, D.C., 6-8 February 1987.

school lists of "obsolete" equipment.³ Some teachers have already suggested to me that computers soon will follow these other technologies into the closets.

The dramatic claims that were made about the revolutionary educational value of earlier technologies are echoed today in claims about alleged benefits of computer-literacy education. Whatever the merit of these claims, once computers are in schools, momentum is established for continuing computer use and upkeep; strong vested interests are at stake here. While acknowledging that promises made about earlier educational technologies were not realized, computer-literacy enthusiasts argue that there is something special about computer technology – it is more interactive, more widespread, more flexible, more complex than the older educational technologies – and that this something special elevates computer technology and makes it essential that schools invest their scarce resources in computers. Strong rhetoric and a variety of pressures unrelated to technology have convinced some teachers that they are shirking their duty to their students, if they do not teach computer literacy. An enormous gap separates the enthusiastic rhetoric – of educators, the media, parents, computer manufacturers, and many computer professionals working in "computers and education" – from the reality of computer-literacy projects in schools.

Computer literacy as a fundamental skill (like reading and writing) is oversold, misapplied, basically trivial in many applications, and not demonstrably up to its claims for education. The founder of an educational software company (and a former educator) put it this way:

"We are still under the influence of a hope more ephemeral than realistic. In fact, the computer has done little that is educationally significant. What it has done is capture our imagination, and prompt us to finance possibly the biggest *unfocused* research effort in the world at a cost, for hardware and software alone, projected to exceed \$8 billion in 1987."⁴

Study and experimentation of new educational ideas is valuable. But when an entire generation of students is the experimental subject, as is the case of computer literacy,

³Krepelka, "Action Plan," Appendix I, 2.

⁴Snyder and Palmer, 12-13.

those who advocate computer-literacy education have a strong obligation to justify their claims. Those who control the educational system have a corresponding duty to evaluate proffered arguments critically.

Computers are tools that, when accompanied by good software such as well designed word processors or spreadsheets, may facilitate mechanical aspects of working with words and numbers. But while computers can help students with some mechanical tasks associated with retrieving data, processing words, formatting text, checking spelling, and revising text, they do not ease the intellectual labor of analyzing information, generating ideas, or expressing thoughts clearly. Contrary to the claims of one computer-literacy teacher, for instance, software such as spelling and grammar checkers will not make us *wiser* human beings.⁵

The computer-literacy courses I learned about are strikingly similar. The construction of these courses defines the "appropriate" level of discourse with respect to computers. What do these courses tell us about "legitimate" computer issues?

By concentrating on low-level computer operation, typical computer-literacy courses *emphasize form at the expense of content*. And by paying little attention to the social context in which computer systems are developed, computer-literacy education restricts the domain of classroom discussions of computers to technical issues. Issues of social priorities generally fall outside this domain and are therefore delegitimized:

"But all forms of computer education, including that for professionals in the field, include a bias against publicly made decisions on how computers ought to be used. Training for computer scientists concentrates on techniques for solving given problems, and bypasses any discussion of which problems are important to solve. Why are so many resources devoted to military applications of computers and comparatively few to medical applications? Why is hardware plentiful enough in the U.S. for substantial amounts of it to be dedicated to game playing, and unavailable in underdeveloped nations for crucial administrative tasks? Questions like these are unasked in our schools, and

⁵Classroom observation.

their absence discourages students from considering the answers important.

There is no public debate on how best to use computing technology.”⁶

By explicitly setting out to “teach” students to “have positive attitudes about computers,” many computer-literacy courses make it difficult for students to develop an informed, critical attitude about computers. Such instruction reflects widespread social enthusiasm about the wonders wrought by science and technology. This enthusiasm with technology, including computer technology, is rarely tempered by a realistic assessment of consequences – realistic enough, for instance, to consider it possible not to proceed with full-scale development and implementation of a new technological application, should the risks involved be too great. Such an assessment of potential risks may not be feasible for a population whose understanding of technology is largely superficial.

By offering only superficial exposure to simplistic models of computation, many computer-literacy courses make it difficult for students to develop a meaningful understanding of computers. Computer-literacy students identify the computer as a very large, fast calculator, but they do not have the faintest conception of it as a symbol-processing device:

“The conceptual essence of the computer as a rule-governed, symbol-manipulating device largely escapes children. ... they seem to have no idea of how the computer transforms that information in the course of its operations. Thus, children lack any solid idea of the computer as a machine which computes.”⁷
[emphasis in the original]

The impoverished models of computing fostered by computer-literacy education render students incapable of accurately understanding computer power. As a result, “many [computer-literacy] students believe in exaggerated capabilities of computers.”⁸ Rather than equipping students to judge for themselves the advantages and risks of proposed

⁶Hank Bromley, “On Computer ‘Literacy’,” class paper, University of Wisconsin, 17 November 1987.

⁷Ronald Mawby, “Structured Interviews on Children’s Conceptions of Computers” (New York: Bank Street College of Education, Center for Children and Technology, February 1984) 36. Technical Report No. 19.

⁸Anderson and Klassen, 144.

new uses of computers, such instruction is more likely to produce a population that will accept the often inflated claims of the computer industry as realistic.

Some computer-literacy enthusiasts argue that computer-literacy education empowers people, by obviating the need for them to depend on computer experts, but many computer-literacy classes actually *encourage* dependence on experts. Students may be told that there are security problems in some computer systems but assured that they need not worry about them, since computer experts are devising ever-better technical solutions to such problems. Similarly, students may be told that there are privacy problems related to the use of computers but that they need not worry about those either, since politicians and other experts have devised laws that solve the problems by “guaranteeing” confidentiality of all files.⁹

Students’ classroom experiences – limited to tiny programs – lead to false models of computing as applied to large, complex, real-world systems. A co-director of one university software-curriculum lab (and a philosophy professor), excited about the “revolutionary” potential of the computer in education, said that “we can use this tiny toy to get the basic principles of computer science across.” He cited four “basic principles” that serve to “demystify the computer”:

1. “A machine can execute cognitive functions (e.g., addition) without understanding what is it doing, by the right programming.
2. With a few primitives, you can build up big machines.
3. With counters, loops, and sense organs, you can build anything.
4. These are all the secrets. The rest is size and speed.”¹⁰

But knowledge about “toy” computers does *not* scale up accurately, to apply to large systems that do real work. In a class discussion of complex computer systems, students extrapolated from their exposure to simple machines and programs to the belief that

⁹Richman, 73.

¹⁰Daniel Dennett, Software, Imagination, Education Conference (Medford, MA: Tufts University, 31 May - 3 June 1988).

computer specialists understand and control complex systems: "But if we create it, someone must understand it."¹¹ While worried about negative impacts of computers, they reassured themselves that there are computer experts out there who are in control of the complicated systems on which we depend. This is comforting but wrong. Many complex computer systems are so large, and have been worked on by so many people, for such long periods of time, that there is no individual or small group of people who understands how the systems work. No one can confidently predict, for example, how such a system will behave in all circumstances nor modify it to fix one "bug" without risking the insertion of another bug. Understanding that many large systems are essentially incomprehensible is critically important to evaluating such systems.

Schools considering computer-literacy instruction need to strip away unsupported claims about it. Educators need to consider first not *how* to teach computer literacy, but *whether* and how computer-literacy courses can contribute to their overall educational goals. The results presented here indicate that computer literacy as it is taught now does not train students for high-level jobs, does not make students "better" thinkers, and does not equip students to make well informed decisions about computer use. A genuine consideration of computer literacy requires that educators be prepared to consider as possible outcomes that computer literacy does not advance their educational goals, or that it does not have a sufficiently high priority to warrant its inclusion in the curriculum. Rejecting the rhetoric about the "computer revolution" or seeing through the "transparent fraudulence" of current computer-literacy education¹² is not Ludditism, it is merely good sense.

Although I reject the unjustified rhetoric surrounding computer-literacy education, I do not think that "computer literacy" must necessarily be jettisoned from all school curricula. Computers affect many important aspects of our lives; it is reasonable that they be discussed at school. The democratic process is best served if the general public has appropriate knowledge to participate in making decisions involving technical systems. The question is, what is appropriate knowledge about computers, and do presently available

¹¹Classroom observation.

¹²Douglas Noble, 64.

computer-literacy programs deliver it? My research leads me to believe that the architects of current computer-literacy courses do not know what computer knowledge the intelligent citizen needs to participate in decisions involving computer systems, and consequently, that current computer-literacy instruction fails to deliver its intended product. It wastes the schools' scarce resources, time and money.

If introductory classes about computing are to be taught, I believe that they should differ greatly from current computer-literacy classes in the following ways: They should spend much less time reviewing the mechanics of operating tiny machines and the syntax of applications software or programming languages. They should spend no time on rote learning of lists, from computer components to uses. They should spend much more time considering the capabilities and limitations of computers and thoughtfully discussing the impacts of important computer applications.

One specific example of material that I believe may contribute to meaningful education about computers is a multi-media presentation entitled "Reliability and Risk: Computers and Nuclear War,"¹³ distributed by Computer Professionals for Social Responsibility, an educational organization. The presentation attacks the myth of computer infallibility by describing different types of computer errors, their sources and consequences. It explains how current political and military trends decrease the time available for people to react to a crisis, thereby shifting critical decision-making responsibilities to computers. It explores the growing reliance on computerized decision-making and how a computer error, especially in times of crisis, could trigger an accidental nuclear war. The program lasts only half an hour and can obviously not cover computer use in critical applications in detail. But it does present salient points about an important and complex area of computer use, greatly heightens people's awareness of problems that they are unlikely to learn about elsewhere, and stimulates exciting discussions and further thought. The presentation uses no computers, and the intended audience need no computer experience.

The point of this presentation, and of other materials that I believe could begin to

¹³Computer Professionals for Social Responsibility and Interlock Media Associates, "Reliability and Risk: Computers and Nuclear War" (Palo Alto, California: CPSR, 1986) 30-minute multi-media presentation.

comprise a meaningful "computer-literacy" course, is to make issues related to computer use clear to a broad audience. This means presenting not only the necessary technical material – some such material is needed, but it need not be too detailed or too complicated for an average audience to understand – but much more importantly, presenting contextual material:

"Real control of the direction the new technology will take involves political understanding, not trivial technical understanding, and it must focus on decisions which affect the design and use of large systems, not on the ability to create catchy little BASIC programs."¹⁴

For instance, consider the 1988 episode in which the U.S.S. Vincennes cruiser shot down an Iranian passenger airplane in the Persian Gulf. Much of the investigation of this incident focused on the ship's computerized sensor-processing system, which is designed to manipulate a huge quantity of raw sensor data into a form that is easily and quickly accessible to people. The investigation concluded that this system had functioned correctly (that is, as intended). This focus on the computer systems provides a relatively narrow view of the incident.

Taking a wider perspective, the investigation also considered another "system," namely the larger system that consists of not only the ship's hardware and software but also of some of its personnel (particularly those involved in operating the intelligence-processing equipment and in using its output to make decisions). The investigation concluded that *this* system had *not* functioned as intended, perhaps because information was incorrectly read from computer displays or incorrectly forwarded to higher-level officers. The investigation felt that the fault for this larger system malfunction lies within its boundaries, and responsibility for the fault was assigned to particular people.

But this episode can be understood better if viewed through an even wider lens. In this case, a wide-angle view reveals machines, people, and the larger context in which the incident took place; i.e., in a politically unstable geographical location, and in the midst of an ongoing military crisis (the ship had been under attack) – all of which combined

¹⁴Douglas Noble, 62.

constitutes a still larger system. With *this* system as contextual framework, we can begin to understand the sorts of problems that can occur when a computer-driven system is applied outside its original design perimeters.

Different kinds of questions are encouraged and discouraged at different levels of abstraction. The advantage of a narrow focus is that it fosters a thorough examination of detail. Such a detailed examination of the Vincennes computer system was certainly relevant. But this advantage has a price. When the view is narrow and the focus is on hardware and software, questions about the nature of the environment in which the technical system is to operate, for example, politics, economics, psychology, and so on, are not asked. Viewed through increasingly wider lenses, nontechnical factors that are often more important than technical factors can come into focus. To understand the Vincennes episode, it is important to know something about politics and crisis management, about sea warfare in restricted as opposed to open waters, etc., and it is relatively unimportant to know something about computer languages. I believe that only a wide-angle view that takes social, economic, and political factors into consideration, can inform the public in a way that may also empower them to judge proposed new computer applications. A wide view is precisely what does not determine current efforts to teach "computer literacy."

Finally, computer-literacy education is a window through which we may learn some general lessons about education. The obsession in some circles with computers-and-education, which often flies in the face of rational evidence, has helped define the level of discourse with respect to education. When the focus of computer-literacy education is on details of hardware and software, questions about larger issues such as teacher preparation and curriculum planning are not raised.

Basic human needs are not being met in many school districts. Teachers work in the front lines of education with insufficient basic resources, space, and equipment of all sorts, inadequate training, and physically decaying facilities. They have little control over the conditions of their working lives, and they have little respect or prestige. Priorities need to be re-ordered when schools are adding computer-literacy classes while graduating significant numbers of students who cannot read. The fundamental problems of an educational system in which 5,200 American public-school teachers are physically attacked

by students each month¹⁵ are unlikely to be solved by a technical fix. The problems are not spurious or random, nor are they being repaired by a technological band-aid; they are systemic.

Educators have suggested a variety of solutions to many of the system's fundamental problems: clarification of schools' goals, upgrading of shoddy school facilities, strengthening networks of community support, enhancing parental involvement, and meeting the health and early education requirements of poor and disadvantaged students, are some suggestions. Teachers would benefit from better reward systems, shared decision-making, significant career-development opportunities, more demanding evaluation methods, significant encouragement of and recognition for creative teaching, and more prestige for their profession. In the hands of a talented, motivated, well supported teacher, virtually any tool, including a computer, may be used effectively. But this presupposes a society that views education as a high priority.

It is tempting to believe that technology can solve tough social problems, human problems. But genuinely wise decision-making about computer use cannot be founded on existing "computer literacy," a conglomeration of bad software, run on outdated hardware, by poorly trained teachers, in unsupportive environments, with vague objectives. While good computers running good software can be convenient tools that facilitate some of the mechanical aspects of school work, there is no justification for mandatory computer-literacy education as it is now taught – unrelated to the promise and problems of large-scale, real-world computing. The student is left with a sense of magic; surely this kind of literacy is a sham. By emphasizing mere recognition, not understanding, computer-literacy classes are turning out people who falsely believe themselves to be computer literate, but who know nothing about real-world computing and its risks.

Social forces have pressed for the use of computers and other new technologies in schools, in the absence of rational evidence that these technologies are useful. These forces include strong market pressures, not only from computer hardware and software manufacturers but also from the professional computers-and-education constituency. The desperation that characterizes the U.S. educational system makes it particularly ripe to

¹⁵Thomas L. Martin Jr., 13.

succumb to promises of quick, glamorous solutions to problems that by now appear intractable.

Computer literacy is hardly a unique example of the prevalence of social pressures over scientific evidence, nor is education uniquely susceptible to such pressures. Often labeled as technological *inevitability*, the process underlying computer-literacy education has been a headlong and uncritical rush toward an exalted technology. If there is one lesson to be learned from computer-literacy education for the next new technology, I hope it is that there is nothing inevitable about this process. Meanwhile, we should not allow the euphoria about computers in schools to become a smokescreen, diverting attention from fundamental educational problems – foremost of which is the lack of *real* literacy among many graduates – that are not solvable by technological means.

6.1 Future Work

Computer literacy may be viewed through different lenses. In this paper, I have tried to present a wider view than is normally taken. This is only a first step in considering the issues that affect computer literacy. Like any interdisciplinary topic, computer-literacy education may be viewed productively from many perspectives, including those of political science, economics, sociology, psychology, history, and ethnography. My work has touched on some of the questions below, any of which would be a starting point to a deeper understanding of computer literacy and the social implications of technology:

- What does computer literacy tell us about the broader issue of technological inevitability? What social forces come into play when a new technology is introduced, how are these forces manipulated, and how have they determined who will control computer use in schools? To what extent does scientific evidence play a role in decisions about new uses of computer technology?
- How do computers in schools serve to amplify long-standing problems? How do they serve as sources of new problems?

- How does the computer industry view education as an economic market? Do different companies have different strategies for exploiting this market?
- How does the introduction of computers differ in a free-market economy as opposed to a Marxian economy?
- How do views of computer-literacy education differ among different education interest groups; e.g., students, teachers, parents, school administrators, state and federal government education departments?
- How does the current content of computer-literacy classes reflect many people's perceptions and fears of computers? Have these perceptions and fears been manipulated? How does it reflect people's perceptions of their roles in society? What does it tell us about people's visions of the future?
- What similarities and differences are revealed by cross-cultural study of computers in schools?

In addition, as mentioned above, I believe that a useful sort of "computer-literacy" curriculum could be developed, by avoiding the superficial, narrow focus of current courses. Work on such a curriculum might bring together stakeholders in different communities, including education and computer science. A "wide-angle" computer-literacy course might consider the risks and benefits of using computers versus the risks and benefits of *not* using computers in different situations. A new course might teach students enough about how computers work to understand how hard it is to completely specify even very simple tasks. It might consider ethical issues associated with computer use; e.g., the ethics of large-scale network use have received a great deal of recent media attention. And it might describe selected applications of computers – important, large-scale, complex applications – from several perspectives, to help students understand the multiplicity of factors that need to be considered.

Chapter 7

Bibliography

Abernathy, Sandra M. and Pettibone, Timothy J. "Computer Literacy and Certification: What States Are Doing." *T.H.E. Journal* (November 1984) 117-119.

Abramson, Gertrude Weinstock. "Adult Computer Competence." *Proceedings 1985 ACM Annual Conference*, 434-438.

Akst, Geoffrey. "Computer Literacy: An Interview with Dr. Michael Hoban." *Journal of Developmental Education*, 8, 2 (1985) 16-19, 31.

Anderson, Ronald E. and Klassen, Daniel L. "A Conceptual Framework for Developing Computer Literacy Instruction." *AEDS Journal*, 4, 3 (Spring 1981) 128-150.

Anderson, Ronald E.; Klassen, Daniel L.; and Johnson, David C. "In Defense of a Comprehensive View of Computer Literacy." Article 8b in Terence R. Cannings and Stephen W. Brown, eds. *The Information Age Classroom: Using the Computer as a Tool*. Irvine, California: Franklin, Beedle & Associates, 1986, 42-44.

"Assessments for Computer Literacy." Dayton, Ohio: Dayton Public School System, 1986.

Austing, Richard H. "A Study of Computer Impact on Society and Computer Literacy Courses and Materials." *Journal of Educational Technology Systems*, 7, 3 (1978-79) 267-274.

Bagnall, Laura D. "The Role of Mental Models in Learning to Program." Class paper for Cognitive Psychology, Massachusetts Institute of Technology (29 July 1988).

Bailey, M. Gene and Tidwell, Rebekah L. "Implementing Computer Literacy: The Year After." *ACM SIGCSE Bulletin*, 18, 1 (February 1986) 24-26. *Proceedings 17th SIGCSE Technical Symposium on Computer Science Education*.

Barbour, Andrew. "Computing in America's Classrooms 1984: The New Computer Literacy Emerges." *Electronic Learning* (October 1984) 39-44.

Baron, Naomi S. "Should Everyone Learn Everything: The Question of Computer Literacy." *SIGCSE Bulletin*, 16, 1 (February 1982) 108-114. *Proceedings 15th SIGCSE Technical Symposium on Computer Science Education*.

Bass, Alison B. "Computers in the Classroom." *Technology Review* (April 1985) 52-62.

Becker, Henry Jay, project director. "Instructional Uses of School Computers: Reports from the 1985 National Survey." Baltimore, Maryland: Center for Social Organization of Schools, The Johns Hopkins University, 1986-87. Series of six newsletters.

_____. "Computers in Schools Today: Some Basic Considerations." *American Journal of Education*, 93, 1 (November 1984) 22-39.

Bergen, Steve and Schalman, Lynne. "Who's Pushing the Buttons?" *Classroom Computer Learning* (December 1984) 52-55.

Berghel, Hal. "Computer Literacy Programs: The Effects of the Large-Class Setting on Student Performance." *Collegiate Microcomputer*, IV, 1 (Spring 1986) 69-81.

Billings, Karen and Moursund, David. **Are You Computer Literate?** Oregon: Dilithium Press, 1979.

Bitter, Gary G. "Computer Literacy Across the Curriculum: A Scope and Sequence Model." *ACM SIGCUE Bulletin*, 17, 4 (Fall 1983) 7-12. [ACM Special Interest Group on Computer Uses in Education.]

Bitter, Gary G. and Craighead, Donna. **Teaching Computer Literacy: Lesson Plans and Activities for Your Classroom (K-4)**. Austin, Texas: Sterling Swift Publishing Company, 1984.

Bitter, Gary G. and Davis, Shelley J. "Measuring the Development of Computer Literacy Among Teachers." *AEDS Journal*, 18, 4 (Summer 1985) 243-253.

Bollerslev, Peter and Estes, Nolan. "Changes in Curricula and Instructional Structures." Paper presented at the International Conference on Education and New Information Technologies (France: Organization for Economic Co-operation and Development (OECD), August 1984).

Bonner, Paul. "Computers in Education: Promise and Reality." Article 5 in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 21-25.

Bork, Alfred. "Let's Test The Power Of Interactive Technology." *Educational Leadership* (March 1986) 36-37.

_____. **Personal Computers for Education**. New York: Harper & Row, 1985.

_____. "Computers and Information Technology as a Learning Aid." In Moonen, J.C.M.M. and Wuite-Harmsma, H.E., eds. **Computers and Education: Policy Problems and Experiences**. New York: North-Holland, 1983. *Proceedings of the Study Conference on Computers and Education* (Enschede, The Netherlands: 18-20 May 1983) 77-90.

Bork, Alfred; Chiocciariello, Augusto; and Franklin, Stephan. "Scientific Reasoning Via The Computer." Irvine, California: Educational Technology Center, Information and Computer Science Department, University of California at Irvine, 4 December 1985. Technical Report.

Brecher, Deborah L. **The Women's Computer Literacy Handbook**. New York: New American Library, 1985.

Bredo, Eric and Feinberg, Walter, eds. **Knowledge and Values in Social and Educational Research**. Philadelphia, PA: Temple University Press, 1982.

Bridwell, L.S.; Sirc, G.; and Brooke, R. "Revising and computers: Case studies of student writers." In Freedman, S., ed. **The Acquisition of written language: Revision and response**. New York: Ablex, 1985.

Bridwell, L.S.; Johnson, P.; and Brehe S. "Computers and composing: Case studies of experienced writers." In Matsuhashi, A., ed. **Writing in real time: Modelling production processes**. New York: Longman, 1983.

Bromley, Hank. "On Computer 'Literacy'." Class paper, University of Wisconsin, 17 November 1987.

Brumbaugh, Kenneth E. "Computer Literacy - 1985." *The Computing Teacher* (1981) 49-50.

Burning Questions. America's Kids: Why They Flunk. ABC News special program (3 October 1988).

"Business Teaching 3 R's to Employees in Effort to Compete." *The New York Times* (Sunday, 1 May 1988) 1:1.

Bybee, Dennis L. "Educational Computing Programs of the Department of Defense Dependents Schools." *T.H.E Journal*, 12, 5 (January 1985) 98-101.

Calfee, Robert. "Computer Literacy and Book Literacy: Parallels and Contrasts." *Educational Researcher* (May 1985) 8-13.

Campbell, Patricia B. and Gulardo, Sonia. "Computers in Education: A Question of Access." *Computers in the Schools*, 1, 1 (Spring 1984) 57-66.

Cannings, Terence R. and Brown, Stephen W. "Learning about Computers: Awareness, Literacy, and Programming." In Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986. Introduction to Section Two, 35-37.

Carey, Max L. "Occupational Employment Growth Through 1990." *Monthly Labor Review*, 104 (August 1981) Table 2.

Center for Learning Technologies, State Education Department. **Computer Literacy: An Introduction**. Albany, New York: The University of the State of New York, 1983.

Chambers, Jack A. and Sprecher, Jerry W. "Computer Assisted Instruction: Current Trends and Critical Issues." *Communications of the ACM*, 23, 6 (June 1980) 332-342.

Chasen, Shelley, and Cohen, Sylvia B. **Basic Computing D, Basic Computing E, and Basic Computing F** (grades 4, 5, and 6). New York, NY: Scholastic Inc., 1984.

Cheng, Tina T.; Plake, Barbara; and Stevens, Dorothy Jo. "A Validation Study of the Computer Literacy Examination: Cognitive Aspect." *AEDS Journal*, 18, 3 (Spring 1985) 139-153.

Cheng, Tina T. and Stevens, Dorothy Jo. "Prioritizing Computer Literacy Topics." *Computers in Education*, 9, 1 (1985) 9-13.

Chorover, Stephan L. "Cautions on Computers in Education." *Byte* (June 1984) 223-226.

Christoffel, Pamela. "Higher Education and New Information Technologies: Trends and Issues." Paper presented at the International Conference on Education and New Information Technologies (France: Organization for Economic Co-operation and Development (OECD), August 1984).

Clarke, V.A. "Sex-Typing of Computing by Primary School Children." Deakin University (1984).

Clements, Douglas H. "Longitudinal Study of the Effects of Logo Programming on Cognitive Abilities and Achievements." *Journal of Educational Computing Research*, 3, 1 (1987) 73-94.

Coburn, Peter, et al. **Practical Guide to Computers in Education**, 2nd Edition. Reading, Massachusetts: Addison-Wesley Publishing Company, 1985.

Cole, Anthony Z. and Riser, Robert R. "Computer Literacy Course for Faculty." In Duncan, Karen and Harris, Diana, eds. **Computers in Education**. New York: IFIP, 1985. *Proceedings IFIP TC3 4th World Conference on Computers in Education - WCCE 85*, 687-690.

"The Commonwealth School of Science and Technology." Draft planning document. Commonwealth of Massachusetts, February 1987.

Compaine, Benjamin. "The New Literacy." *Science Digest* (March 1983) 16-19+.

"Computer Education Curriculum: Kindergarten-Grade 6." Arlington, Massachusetts: Arlington Public School System, September 1985. Draft.

Computer Literacy: Elementary Grades and Computer Literacy: Intermediate and Secondary Grades. Brooklyn, New York: Board of Education, City of New York, 1983.

"Computer Literacy Course of Study" and "Assessments for Computer Literacy." Dayton, Ohio: Dayton Public School System, 1986.

"Computer Literacy Curriculum Guide: 1984-1985." Amherst, Massachusetts: Amherst Central School District.

"Computer Literacy Objectives from MECC" (Minnesota Educational Computing Consortium). *ACM SIGCUE Bulletin*, 13, 4 (October 1979) 29-34. [ACM Special Interest Group on Computer Uses in Education.]

Computer Professionals for Social Responsibility and Interlock Media Associates. "Reliability and Risk: Computers and Nuclear War." Palo Alto, California: CPSR, 1986. Half-hour multi-media presentation.

"Computers for Intellectual Regeneration." *Social Education* (January 1987) 34-36. Interview with Professor Robert Taylor.

"Computers Go To School." *Industry Week* (22 July 1985) 39-40.

"Computers in the Curriculum: A Computer Literacy Guide." MECC [Minnesota Educational Computing Consortium], December 1985.

Corcoran, Thomas B.; Walker, Lisa J.; and White, J. Lynne. "Working in Urban Schools." Washington, D.C.: Institute for Educational Leadership, 1988.

Cuffaro, Harriet K. "Microcomputers in Education: Why Is Earlier Better?" *Teachers College Record*, 85, 4 (Summer 1984) 559-568.

"Cupertino School District Develops Computer Literacy Curriculum." *The Computing Teacher* (1981) 27-34.

Curran, Susan and Curnow, Ray. **Overcoming Computer Illiteracy: A Friendly Introduction to Computers.** New York: Penguin Books, Inc., 1984.

Daiute, Colette. **Writing & Computers.** Reading, Massachusetts: Addison-Wesley Publishing Company, 1985.

Dalby, John; Tourniaire, Françoise; and Linn, Marcia C. "Making Programming Instruction Cognitively Demanding: An Intervention Study." Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana.

Davidson, Jan. "Computers are Good for Kids ... and Families." *Computer Learning Month* (Software Publishers Association, October 1987) 6-7.

Dede, Christopher. "New Technology and Education." **Encyclopedia of Educational Communications Media Technology**, 2nd edition (Winter 1984). Pre-publication copy.

DeLoughry, Thomas J. "Failure of Colleges to Teach Computer Ethics is Called Oversight with Potentially Catastrophic Consequences." *The Chronicle of Higher Education* (24 February 1988) A:15 and A:18.

Devon, Richard F. "In Praise of Computer Illiteracy." *Bulletin of Science, Technology & Society*, 7, 1 & 2 (1987) 338-343. Paper presented at the Second National Conference on Technological Literacy (Washington, D.C.: 6-8 February 1987).

diSessa, Andrea A. "Artificial Worlds and Real Experience." *Instructional Science*, 14 (1986) 207-227.

Doyle, Barbara. "Should a Computer Literacy Class be the First Required Course for Data Processing Majors?" *SIGCSE Bulletin*, 18, 2 (June 1986) 10-12. [ACM Special Interest Group on Computer Science Education.]

Dudley-Marling, Curt and Owston, Ronald D. "The State of Educational Software: A Criterion-Based Evaluation." *Educational Technology*, XXVII, 3 (March 1987) 25-29.

"EDCO Sample Elementary Computer Curriculum Objectives." Newton, Massachusetts: 1985.

EDUCOM Computer Literacy Newsletter, 1, 1 (Fall 1983); 1, 2 (Winter 1984); 1, 3 (Spring 1984); 2, 1 (Fall 1984); 2, 2 (Winter 1985); 2, 3 (Spring 1985); 2, 4 (Fall 1985).

Ehrlich, Kate, et al. "Issues and Problems in Studying Transfer Effects of Programming." Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana.

Ekins, Roger Robin. "Computer Illiterate, and Proud of It." *AACJC Journal* (October/November 1985) 28-31.

Elder, Carlyn. "Development of a Computerized World Geography Database with Accompanying Documentation to Enhance Information-Processing Skills and Geography Knowledge." Presented at the Third National Conference on Technological Literacy (Arlington, Virginia: 5-7 February 1988).

Ellison, Robert J. "Microcomputers and Computer Literacy: A Case Study." *Proceedings NECC/2: National Educational Computing Conference* (1980) 68-72.

Feigenbaum, Edward A. and McCorduck, Pamela. **The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World**. Reading, Massachusetts: Addison-Wesley Publishing Company, 1983.

Fischer, Herman. "Computer Literacy Scope and Sequence Models: A Critical Review of Two Approaches." *ACM SIGCUE Bulletin*, 18, 1 (Winter 1984) 22-28. [ACM Special Interest Group on Computer Uses in Education.]

Fisher, Lawrence M. "PC Makers' Campaign in Schools." *The New York Times* (19 July 1988) D:1ff.

Fiske, Edward B. "Computers, in Most Schools, Have Brought No Revolution." *The New York Times* (Sunday, 9 December 1984) 1:1ff.

_____. "School Computers: Vision and Reality." *The New York Times* (11 December 1984) B:2ff.

_____. "Technology Gaining a Place in Liberal Arts Curriculums." *The New York Times* (Sunday, 29 April 1984) 1:1-2.

Franklin, Stanley P. "A Liberal Arts Computer Literacy Course: A Preliminary Report." *Proceedings 1982 National Educational Computer Conference (NECC '82)* (University of Missouri) 120-122.

Fritz, Jane M. "Rethinking Computer Literacy." In Duncan, Karen and Harris, Diana, eds. **Computers in Education**. New York: IFIP, 1985. *Proceedings IFIP TC3 4th World Conference on Computers in Education - WCCE 85*, 705-710.

Gabriel, Roy M. "Assessing Computer Literacy: A Validated Instrument and Empirical Results." *AEDS Journal*, 18, 3 (Spring 1985) 153-171.

_____. "Computer Literacy Assessment and Validation: Empirical Relationships at Both Student and School Levels." *Journal of Educational Computing Research*, 1, 1 (1985) 415-425.

Gilbert, Steven W. "The Conversion Experience." *The Software Initiative Newsletter*, 2, 2 (May/June 1988) 5. Published by EDUCOM.

Gilmore, Perry and Glatthorn, Allan A., eds. **Children In and Out of School: Ethnography and Education**. Washington, D.C.: Center for Applied Linguistics, 1982.

Gonzalez, Marianne. "Elementary Computer Curriculum Retrospective." Acton, Massachusetts: Acton Public School System, 12 November 1985.

Goodman, Frederick L. "Computers and the Future of Literacy." *Proceedings 1981 National Computer Conference* (Virginia: AFIPS Press, 1981) 601-603.

Goodson, Mrs. Bobby. "Computer Literacy in the Elementary School: A District-Wide Program." *Proceedings 1982 National Educational Computer Conference (NECC '82)* (University of Missouri) 235-237.

Gorman, Christine. "The Literacy Gap." *TIME* (19 December 1988) 56-57.

Gwyn, Rhys. "New Teaching Functions and Implications for New Training Programmes." Paper presented at the International Conference on Education and New Information Technologies (France: Organization for Economic Co-operation and Development (OECD), August 1984).

- Hamblen, John W. "Computer Literacy and Societal Impact of Computers: Education and Manpower." *Computers & Society*, 11, 3 (Summer 1981) 19-22.
- Hanavan, Patricia Relf. **Basic Computing A** and **Basic Computing B** (grades 1 and 2). New York, NY: Scholastic Inc., 1984.
- Hanavan, Patricia Relf; Chasen, Shelley; and Cohen, Sylvia B. **Basic Computing C** (grade 3). New York, NY: Scholastic Inc., 1984.
- Hansen, Thomas P., et al. "Computer Literacy Courses." *AEDS Monitor* (November 1979).
- Harvey, Brian. "Stop Saying 'Computer Literacy'!" Article 9 in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 45-48. Reprinted from *Classroom Computer News* (May/June 1983).
- Hawkins, Jan. "Computers and Girls: Rethinking the Issues." New York: Bank Street College of Education, Center for Children and Technology, April 1984. Technical Report No. 24.
- Hawley, David. **Computer Literacy**. Oregon: Oregon School Study Council, 1984. Printed as *OSSC Bulletin*, 27, 8 (April 1984).
- Hechinger, Fred M. "What Illiteracy Isn't." *The New York Times* (2 June 1987) C:7.
- Hertlein, Professor Grace C. "Computer Literacy 1982: An Update." *Proceedings 1982 National Educational Computer Conference* (University of Missouri) 238-244.
- Holden, Constance. "Will Home Computers Transform Schools?" *Science*, 225, p. 296.
- Hopper, Judith A. and Bruckhard, Glenn. "Computer Literacy for Teachers: A District Do-It-Yourself Kit." *Proceedings 1982 National Educational Computer Conference* (University of Missouri) 323-325.
- Horn, Carin E. and Poirot, James L. **Computer Literacy: Problem-Solving with Computers**, second edition. Austin, Texas: Sterling Swift Publishing Company, 1985.
- Horton, Forest Woody, Jr. "Information Literacy vs. Computer Literacy." *ASIS Bulletin* (April 1983) 14-16. [American Society for Information Science.]
- Illich, Ivan. "Computer Literacy and the Cybernetic Dream." *Bulletin of Science, Technology & Society*, 7, 1 & 2 (1987) 306-309. Paper presented at the Second National Conference on Technological Literacy (Washington, D.C.: 6-8 February 1987).
- . "Silence is a Commons." *The CoEvolution Quarterly* (Winter 1983) 5-9.
- International Council for Computers in Education. **Computer Literacy Activities for Elementary & Middle School Students**. Oregon: ICCE, 1984.

Jackson, Colonel Thames F., Jr. and McKee, Major Timothy C. "Computer Literacy: Practical Necessity." *Proceedings 1982 National Educational Computer Conference* (University of Missouri) 155-160.

Johnson, D.C. and Anderson, R.E. "Algorithms and Programming as Part of Computer Literacy - Some Assessment Results With Implications for Secondary School Mathematics." In Duncan, Karen and Harris, Diana, eds. **Computers in Education**. New York: IFIP, 1985. *Proceedings IFIP TC3 4th World Conference on Computers in Education - WCCE 85*, 691-698.

Johnson, David C., et al. "Computer Literacy - What Is It?" *Mathematics Teacher*, 73, 2 (February 1980) 91-96.

Kay, Alan. "Computer Software." *Scientific American*, 251, 3 (September 1984) 53-59.

Kelman, Peter. "Computer Literacy: A Critical Re-Examination." *Computers in the Schools*, 1, 2 (Summer 1984) 3-18.

Klassen, Daniel. "Computer Literacy Revisited." *AEDS Journal*, 16, 1+2 (Fall / Winter 1983) 41-48.

Kozol, Jonathan. **Illiterate America**. New York: New American Library, 1985.

Krause, Kenneth L. "The Delivery of Education Using Current Technology." *Proceedings 1985 ACM Annual Conference*, 342-349. Panel discussion.

Krepelka, Joanne. "Action Plan for Computer Technology, 1986-87." Cambridge, Massachusetts: Cambridge Public School System, 1986. Computer coordinator's report.

———. "Progress Report on Action Plan, 1986-87." Cambridge, Massachusetts: Cambridge Public School System, 1987. Computer coordinator's report.

Kuhlman, Edward. "Cipherring of the School: Computer Litter-acy." *Clearing House* (September 1984).

Kurland, D. Midian, et al. "A Study of the Development of Programming Ability and Thinking Skills in High School Students." *Journal of Educational Computing Research*, 2, 4 (1986).

Kurland, D. Midian; et al. "Mapping the Cognitive Demands of Learning to Program." Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana.

Kurland, D. Midian; Mawby, Ronald; and Cahir, Nancy. "The Development of Programming Expertise in Adults and Children." Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana.

Lariccia, Giovanni and Megarry, Jacquetta. "Information Society and General Education." Paper presented at the International Conference on Education and New Information Technologies (France: Organization for Economic Co-operation and Development (OECD), August 1984).

Laudon, Kenneth C. **Dossier Society: Value Choices in the Design of National Information Systems**. New York: Columbia University Press, 1986.

Layman, Donald L. "Information Literacy and Educational Productivity." *Proceedings 46th Annual ASIS Meeting* (1983) 309-312. [American Society for Information Science.]

Levin, Henry M. and Rumberger, Russell W. "The Educational Implications of High Technology." Stanford, California: Institute for Research on Educational Finance and Governance, School of Education, Stanford University, February 1983. Project Report No. 83-A4.

Levine, Art. "Something doesn't quite compute." *U.S. News & World Report*, 101, 19 (10 November 1986) 78-79.

Lewis, Peter H. "The Computer Revolution Revised." *The New York Times* (Sunday, 7 August 1988) 4A:15-17.

Linn, Marcia C. "Gender Equity in Computer Learning Environments." *Computers and the Social Sciences*, 1 (1985) 19-27.

Lockheed, Marlaine E.; Nielsen, Antonia; and Stone, Meredith K. "Determinants of Microcomputer Literacy in High School Students." *Journal of Educational Computing Research*, 1, 1 (1985) 81-96.

Lombardi, John V. **Computer Literacy: The Basic Concepts and Language**. Indiana: Indiana University Press, 1983.

Longstreet, Wilma S. and Sorant, Peter E. "Computer Literacy - Definition?" *Educational Horizons* (Spring 1985) 118-120.

Loop, Liza; Anton, Julie; and Zamora, Ramon. **Computertown: Bringing Computer Literacy to Your Community**. Reston, Virginia: Reston Publishing Company, 1983.

Luehrmann, Arthur. "The Best Way to Teach Computer Literacy." Article 10a in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 49-52.

_____. **Computer Literacy: A Hands-On Approach**. U.S.A.: McGraw-Hill, Inc., 1983.

_____. "Computer Literacy - What Should It Be?" *Mathematics Teacher*, 74, 9 (December 1981) 682-686.

- Luehrmann, Arthur and Peckham, Arthur. **Computer Literacy: A Hands-On Approach, First Apple Edition.** U.S.A.: McGraw-Hill, Inc., 1983.
- Macroff, Gene I. "The Real Job Boom Is Likely to be Low-Tech." *The New York Times* (4 September 1983) E:16.
- Many, Wesley A.; et al. "The Effect of Learning to Program in Logo on Reasoning Skills of Junior High School Students." *Journal of Educational Computing Research*, 4, 2 (1988) 203-213.
- Manzelli, Jane; Brown, Elizabeth; and Morse, Frances. "Brookline Public School System: Computer Education Long Range Goals and Implementation Plan, 1986-1991." Brookline, Massachusetts: 30 May 1986.
- Martin, Jean Buddington and Martin, Kenneth E. "A Profile of Today's Computer Literacy Student." *ACM SIGCSE Bulletin*, 18, 1 (February 1986) 27-33. *Proceedings 17th SIGCSE Technical Symposium on Computer Science Education.*
- Martin, Thomas L., Jr. "Intellectual Capital Formation and the Education System." National Academy of Engineering, *The Bridge*, 17, 3 (Fall 1987) 10-14.
- Masat, Francis E. **Computer Literacy in Higher Education.** Washington, D.C.: American Association for Higher Education, 1981. AAHE-ERIC/Higher Education Research Report No. 6.
- Matsumoto, Carolee. "The Potential of Computers for Teaching Thinking." Article I in Cannings, Terence R. and Brown, Stephen W., eds. **Telecommunications, Robots, and Problem Solving.** Irvine, California: Franklin, Beedle & Associates, 1987, 35-39.
- Mawby, Ronald, et al. "Structured Interviews on Children's Conceptions of Computers." New York: Bank Street College of Education, Center for Children and Technology, February 1984. Technical Report No. 19.
- Mayer, Richard E.; Dyck, Jennifer L.; and Vilberg, William. "Learning to Program and Learning to Think: What's the Connection?" *Communications of the ACM*, 29, 7 (July 1986) 605-610.
- McCracken, Daniel D. "A Skeptical View of Computer Literacy." *SIGCSE Bulletin*, 16, 1 (February 1984) Appendix A. *Proceedings 15th ACM SIGCSE Technical Symposium on Computer Science Education.*
- Menosky, Joseph A. "Computer Worship." *Science '84* (May 1984) 40-46.
- "Mission: Define Computer Literacy. The Illinois-Wisconsin ISACS Computer Coordinators' Committee on Computer Literacy Report (May 1985)." *The Computing Teacher* (November 1985) 10-15.
- Molnar, Andrew R. "The Next Great Crisis in American Education: Computer Literacy." *Journal of Educational Technology Systems*, 7, 3 (1978-79) 275-284.

Morse, Frances and Roth, Michael. "Grade 5 Logo: Guidelines & Ideas." Brookline, Massachusetts: Brookline Public School System, August 1986.

Morse, Frances and Tieman, Joan. "A Logo Handbook, Second Edition." Brookline, Massachusetts: Brookline Public School System, August 1985. Grade 4 curriculum.

Moursund, David. **Precollege Computer Literacy: A Personal Computing Approach.** Eugene, Oregon: International Council for Computers in Education, 1981.

Naiman, Adeline. "A Hard Look at Educational Software." *Byte* (February 1987) 193-200.

National Assessment of Educational Progress. "Computer Competence: The First National Assessment." Princeton, New Jersey: Educational Testing Service, 1988.

National Commission on Excellence in Education. "A Nation at Risk: The Imperative for Education Reform." Washington, D.C.: U.S. Government Printing Office, April 1983.

Neibauer, Alan. "The Computer Literacy Myth." *T.H.E. Journal* (February 1985) 88-90.

Nielsen, Janni. "Not the Computer but Human Interaction is the Basis for Cognitive Development and Education." Accepted for publication, 1986.

Niemiec, Richard and Walberg, Herbert J. "Comparative Effects of Computer-Assisted Instruction: A Synthesis of Reviews." *Journal of Educational Computing Research*, 3, 1 (1987) 19-37.

Noble, Douglas. "The Underside of Computer Literacy." *Raritan* (Spring 1984) 37-64.

Organization for Economic Co-operation and Development (OECD), Secretariat. "The Introduction of the New Information Technologies in Education: Policy Trends and Developments in Member Countries." Paper presented at the International Conference on Education and New Information Technologies (France: August 1984).

Pea, Roy D. "Socializing the Knowledge Transfer Problem." *International Journal of Educational Research*, 11, 6 (1987) 639-663.

——— "Beyond Amplification: Using the Computer to Reorganize Mental Functioning." *Educational Psychologist*, 20, 4 (1985) 167-182.

——— "Language-Independent Conceptual 'Bugs' in Novice Programming." New York: Bank Street College of Education, Center for Children and Technology, December 1984. Technical Report No. 31.

——— "What Will It Take to Learn Thinking Skills Through Computer Programming?" Symposium on Development Studies of Computer Programming Skills, 1984 AERA Annual Meeting (American Educational Research Association), New Orleans, Louisiana.

_____. "Prospects and Challenges for Using Microcomputers in Schools." New York: Bank Street College of Education, Center for Children and Technology, February 1984. Technical Report No. 7.

_____. "Logo Programming and Problem Solving." Symposium on Chameleon in the Classroom: Developing Roles for Computers, 1983 AERA Annual Meeting (American Educational Research Association), Montreal, Canada.

Pea, Roy D. and Kurland, D. Midian. "Logo Programming and the Development of Planning Skills." New York: Bank Street College of Education, Center for Children and Technology, March 1984. Technical Report No. 16.

_____. "On the Cognitive Effects of Learning Computer Programming: A Critical Look." New York: Bank Street College of Education, Center for Children and Technology, January 1984. Technical Report No. 9.

Pearlman, Robert; Weizenbaum, Joseph; and Papert, Seymour. "Computers and Education." Talks given in a series on Computers and Society, Massachusetts Institute of Technology, 19 March 1985.

Penrod, James and Hughes, Nancy. "Computer Literacy: Synopsis Of An Institute." *Cause/Effect* (November 1983) 3-7.

Pipho, Chris. "The Computer Literacy Dilemma in the Public Schools." *Educational Horizons* (Spring 1985) 100-103.

Poyner, Rick. "New Toy: An Inquiry into Educational Computing." B.S. Thesis, Computer Science Department, Massachusetts Institute of Technology (May 1987).

"QUEUE: The Best in Educational Software." Software catalogue. Bridgeport, CT: QUEUE, Inc., 1987.

Quist, Anton Braun. "'Computer Literacy': Wave of the Future - Or Relic of the Past?" *Computers in the Schools*, 1, 1 (Spring 1984) 67-78.

Rabinow, Paul and Sullivan, William M., eds. **Interpretive Social Science: A Reader**. California: University of California Press, 1979.

Radin, Stephen and Greenberg, Harold M. **Computer Literacy for School Administrators & Supervisors**. Lexington, MA: Lexington Books, 1983.

Randhawa, Bikkar S. and Hunt, Dennis. "Computers and Computer Literacy in Contemporary Psychological, Socio-Economic, and Educational Context." *AEDS Journal*, 17, 3 (Spring 1984) 1-13.

"Reader's Survey Results: What Is Computer Literacy?" *Classroom Computer Learning* (March 1986) p. 53.

Resnick, Daniel P. and Resnick, Lauren B. "The Nature of Literacy: An Historical Exploration." *Harvard Educational Review*, 47, 5 (August 1977) 370-385.

Richman, Ellen. **Spotlight on Computer Literacy**. Framingham, Massachusetts: Random House School Division, 1982.

Ridley, William J. and Hull, McAllister, H., Jr. "Transforming American Education: Reducing the Risk to the Nation. A Report to the Secretary of Education, United States Department of Education, by the National Task Force on Educational Technology." Washington, D.C.: U.S. Department of Education, Office of Educational Research and Improvement, April 1986.

Ries, Daniel R., et al. "The Role of Computers in Belmont Public Schools: Final Report of the Advisory Committee to the School Committee on the Use of Computers." Belmont, Massachusetts: February 1984.

Roberts, Jack L. **Scholastic Computing: An Introduction to Computers**. Teacher's Edition. New York, NY: Scholastic Inc., 1984.

———. **Understanding Computers**. New York, NY: Scholastic Inc., 1984.

Rosenberg, Ronni L. "Computer Literacy Education." *Bulletin of Science, Technology, and Society*, special issue on technological literacy (to appear, Fall 1988).

———. "Computer Literacy in Schools." Symposium on Directions and Implications of Advanced Computing (DIAC-88) (St. Paul, Minnesota: 21 August 1988).

———. "Computer Literacy: A Study of Primary and Secondary Schools." Third National Conference on Technological Literacy (Arlington, Virginia: 5-7 February 1988).

———. "A Critical Analysis of Research on Intelligent Tutoring Systems." *Educational Technology* (November 1987) 7-13.

———. "Computers and Writing." Class paper for Ethnographic Research in Educational and Community Settings, Harvard University (May 1985).

Rosenthal, Elisabeth. "A Rebel in the Computer Revolution." *Science Digest* (August 1983) 94-97. Interview with Joseph Weizenbaum.

Roszak, Theodore. **The Cult of Information: The Folklore of Computers and the True Art of Thinking**. New York: Pantheon Books, 1986.

Ryder, Barbara A. "A 'Hands-On' Approach to Computer Literacy." *SIGCSE Bulletin*, 16, 1 (February 1984) 102-107. *Proceedings 15th ACM SIGCSE Technical Symposium on Computer Science Education*.

Salomon, Gavriel. "AI in Reverse: Computer Tools that Turn Cognitive." *Journal of Educational Computing Research*, 4, 2 (1988) 123-139.

Salomon, Gavriel and Perkins, D.N. "Transfer of Cognitive Skills From Programming: When and How?" *Journal of Educational Computing Research*, 3, 2 (1987) 149-169.

Salpeter, Judy. "Interactive: The Truth Behind the Promises." *Classroom Computer Learning* (November/December 1986) 26-34.

Schiffman, Shirl S. "Designing a Computer Literacy Curriculum: An Integrated Approach." In Duncan, Karen and Harris, Diana, eds. **Computers in Education**. New York: IFIP, 1985. *Proceedings IFIP TC3 4th World Conference on Computers in Education - WCCE 85*, 677-680.

Schloss, Patrick J.; Wisniewski, Lech A.; and Cartwright, G. Phillip. "The Differential Effect of Learner Control and Feedback in College Students' Performance on CAI Modules." *Journal of Educational Computing Research*, 4, 2 (1988) 141-150.

Schwartz, Judah. "Software to Think With." Talk presented at CPSR/Boston meeting (Computer Professionals for Social Responsibility) 2/17/88.

Schwartz, Judah and Yerushalmy, Micchal. "The Geometric Supposer: A Case Study in the Use of Microcomputers in Mathematics Education." Educational Technology Center, Harvard University, December 1984. Technical paper.

Scrogan, Len. "Telecomputing: How to Overcome the Roadblocks." *Classroom Computer Learning* (February 1987) 40-45.

Seidel, Robert J.; Anderson, Ronald E.; and Hunter, Beverly; eds. **Computer Literacy: Issues and Directions for 1985**. New York: Academic Press, 1982. New York: Academic Press, Inc., 1982. *Proceedings of the National Computer Literacy Goals for 1985 Conference* (Reston, Virginia: 18-20 December 1980).

Serrin, W. "Computers divide AT&T and its workers." *New York Times* (18 November 1983) 1.

Severo, Richard. "Computer Makers Find Rich Market in Schools." *The New York Times* (10 December 1984) B:1ff.

Sheingold, Karen; Hawkins, Jan; and Char, Cynthia. "'I'm the Thinkist, You're the Typist': The Interaction of Technology and the Social Life of Classrooms." New York: Bank Street College of Education, Center for Children and Technology, June 1984. Technical Report No. 27.

Shore, John. **The Sachertorte Algorithm and Other Antidotes to Computer Anxiety**. New York: Penguin Books, 1985.

———. "Computers at Capital Hill Day School." Washington, D.C.: 23 September 1982.

Smith, D.J. and Sage, M.W. "Computer Literacy and the Education/Training Interface." *T.H.E. Journal*, 7, 4 (1983) 227-234.

Snyder, Tom and Palmer, Jane. **In Search of the Most Amazing Thing: Children, Education, and Computers**. Reading, Massachusetts: Addison-Wesley Publishing Company, 1986.

Solomon, Cynthia. **Computer Environments for Children**. Cambridge, Massachusetts: The MIT Press, 1986.

Soloway, Elliot. "Teaching Synthesis Skills Via Programming." Talk presented at MIT Arts and Media Technology Laboratory, 12 November 1987.

Spain, T. "How State Governments are Helping to Define Computer Literacy." Article 10b in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 53-55.

Spero, Samuel. "Let Them Have Micros: An Approach to Computer Literacy for Teachers." *T.H.E. Journal*, 10, 1 (September 1982) 127-129.

Stephenson, Bette (Minister of Education, Ontario, Canada). "General Policies in the Introduction of New Information Technologies in Education." Paper presented at the International Conference on Education and New Information Technologies (France: Organization for Economic Co-operation and Development (OECD), August 1984).

Stern, Nancy and Stern, Robert A. **Computers in Society**. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1983.

Stuckey, John. "The Computing Skills Workshop at Carnegie-Mellon University." *Outlook*, 19, 1/2 (Spring/Summer 1986) 46-48. [Publication of ACM SIGCUE, Special Interest Group on Computer Uses in Education.]

Stuckman, Ralph. "A Model to Develop Computer Literacy in Education." In Duncan, Karen and Harris, Diana, eds. **Computers in Education**. New York: IFIP, 1985. *Proceedings IFIP TC3 4th World Conference on Computers in Education - WCCE 85*, 681-685.

"Study Finds Children Know Computer's Parts, Not Whole." *The New York Times* (7 April 1988) A:24.

"Sunburst Educational Computing Courseware." Software catalogue. Pleasantville, NY: Sunburst Communications, 1987.

Tift, Susan. "Who's Teaching Our Children?" *TIME*, 132, 20 (14 November 1988).

Trainor, Timothy N. **Computer Literacy: Concepts and Applications**. Santa Cruz, California: Mitchell Publishing, Inc., 1984.

Tucker, Marc S. "Computers in the Schools: What Revolution?" Article 21 in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 102-107.

"Unmasking some myths about computer literacy." *Modern Office Technology* (June 1984) 116+.

U.S. Bureau of Labor Statistics. "Occupational Outlook Handbook, 1982-1983." Washington, D.C.: Superintendent of Documents, U.S. Government Printing Office, Bulletin No. 2200.

U.S. Congress, Office of Technology Assessment. "Federal Government Information Technology: Electronic Record Systems and Individual Privacy." OTA-CIT-296. Washington, D.C.: U.S. Government Printing Office, June 1986.

Van Dyke, Carolynn. "Taking 'Computer Literacy' Literally." *Communications of the ACM*, 30, 5 (May 1987) 366-374.

Van Maanen, John. **Tales of the Field: On Writing Ethnography**. Chicago, IL: The University of Chicago Press, 1988.

Vigil, Peter J. "Computer Literacy and the Two Cultures Revisited." *Proceedings 48th Annual ASIS Meeting* (1985) 240-242. [American Society for Information Science.]

Vockell, Edward L.; Rivers, Robert H.; and Kozubal, Diane. "Computer Literacy for Teachers: An Intensive Program." *Proceedings 1982 National Educational Computer Conference (NECC '82)* (University of Missouri) 326-330.

Watt, Daniel. "Should Children Be Computer Programmers?" *Educational Computing* (September 1982) 130-133.

———. "Logo Curriculum Materials Specially Adapted for the Brookline Public Schools." Brookline, Massachusetts: June 1981.

Weiss, Eric A. "Two Contributions to Computer Literacy." *Abacus*, 3, 2 (Winter 1986) 46-51.

———. "Computer Science and Literacy Defined." *Abacus*, 3, 3 (Spring 1986) 48-51.

Weizenbaum, Joseph. "Computers and Education: The Computer in Schools." Talk presented at the 1986 Lustrum Congress (Holland: Technische Hogeschool Twente, November 1986).

Wesley, Beth Eddinger; Krockover, Gerald H.; and Hicks, Charles R. "Locus of Control and Acquisition of Computer Literacy." *Journal of Computer-Based Information*, 12, 1 (Winter 1985) 12-16.

"What is Computer Literacy?" Article 10c in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 56-57.

Williams, Dennis A. "How One School Does It Right." Article 20 in Cannings, Terence R. and Brown, Stephen W., eds. **The Information Age Classroom: Using the Computer as a Tool**. Irvine, California: Franklin, Beedle & Associates, 1986, 92-96.

Williams, Harvey. "Computer Literacy for Teachers in Manitoba." *Computers in Education*, 8, 4 (1984) 393-395.

Willis, Jerry. "Educational Computing: Current Status and Future Directions." *Computers in the Schools*, 1, 1 (Spring 1984) 3-12.

Wilson, David S. "Computer's Role in Class Expands." *The New York Times* (13 April 1988) B:11.

Wilson, J.D. and Trenary, R.G. "An Investigation of Computer Literacy as a Function of Attitude." *SIGCSE Bulletin*, 13, 3 (September 1981) 5-12. [ACM Special Interest Group on Computer Science Education.]

Winner, Alice-Ann. "Computer Literacy in the Elementary School: An Argument for Change from Within." *AEDS Journal*, 16, 3 (Spring 1983) 153-165.

Winner, Langdon. **The Whale and the Reactor**. Chicago: University of Press, 1986.

Woodson, M.I. Charles E. "Computer Literacy by Computer." *SIGCSE Bulletin*, 14, 1 (February 1982) 226-228. *Proceedings 13th SIGCSE Technical Symposium on Computer Science Education*.

Yang, Jung-Shing. "Individualizing Instruction Through Intelligent Computer-Assisted Instruction: A Perspective." *Educational Technology*, XXVII, 3 (March 1987) 7-15.

Zamora, Ramon. "Computer and Other Literacies." Chapter 2 in Grady, M. Tim and Gawronski, Jane D., eds. **Computers in Curriculum and Instruction**. Association for Supervision and Curriculum Development, 1983.

Appendix A

Survey On Computer Literacy (for Teachers)

- Please write as much as you like, and continue any answers on the back or on other sheets.
- If you are teaching now:
 - What do you teach, at what level, and where?
 - Does your school have computers? If so, what type(s) and how many?
 - Does your school plan to buy more hardware? software?
 - If your school or district has a computer specialist, what is this person's background and function?
 - Do you have previous experience with computers?
 - If your school has a computer requirement, what is required?
 - If your school teaches Computer Literacy, at what level(s) is it offered, who teaches it, and what material is covered?
- If you aren't teaching now, what do you plan to do after attending this class?
- Why are you taking this course? As a requirement for a degree (which one)? For personal reasons? Both?
- How do you define Computer Literacy?

- Do you think Computer Literacy should be a school requirement? Why or why not?
- Whatever your answers to the question above, I'm very interested in your underlying reasons. If you think Computer Literacy encourages certain qualities, why? If you think it doesn't, why not?
- If you think Computer Literacy should be required:
 - What is/are the right level(s), and how long a class should it be?
 - What should be taught?
 - How would you fit it into the class schedule? Would something else have to be dropped?
 - How would you fit it into the school budget?
- If you have doubts about requiring Computer Literacy, please explain them.
- What teacher training in computer literacy do you think is appropriate? How does this compare with the training of the computer teachers you know?
- How does computer literacy serve what you think are the fundamental goals of the educational system (e.g., the list of things that should be taught in school, which you handed in awhile ago)?
- You already know that there is a lot of interest in Computer Literacy in schools. What do you think is behind all this interest – how do you explain it? Do you think it is justified or not?

Appendix B

Survey On Computer Literacy (for Computer Professionals)

The following messages were distributed in several issues of the RISKS-LIST, an electronic digest whose full name is "Forum on Risks to the Public in Computers and Related Systems." RISKS is moderated by Peter G. Neumann, representing the ACM (Association for Computing Machinery) Committee on Computers and Public Policy. Since RISKS's inception in August 1985, an average of 170 issues per year have been distributed; each issue contains approximately eight messages. RISKS reaches a very wide variety of computer networks worldwide (including the United States -- through the Internet and many local bulletin boards, western Europe, Japan, Australia, and New Zealand). In addition, highlights of RISKS are printed in the ACM's *Software Engineering Notes*, which has a readership of 12,000 people, many of whom do not read the on-line version of RISKS. While it is very difficult to estimate the total readership of RISKS at any time, Peter Neumann's rough estimate is on the order of 100,000 people. To subscribe to RISKS, send requests to the Internet address risks-request@csl.sri.com.

RISKS-LIST: RISKS-FORUM Digest Tuesday 25 October 1988 Volume 7 : Issue 67

Date: Tue, 25 Oct 88 10:36:36 edt
From: Ronni Rosenberg {ronni@VX.LCS.MIT.EDU}
Subject: Computer Literacy

I am writing a Ph.D. thesis on computer-literacy education. One way in which this work differs from previous work is that it incorporates the perspectives of not only educators, but also computer professionals, the most computer-literate group in society. (To the extent that "computer literacy" means anything, it must apply to computer professionals.) To get more feedback from the computer community, I am starting a RISKS dialogue on computer literacy.

I will be sending several messages about computer literacy, asking for your opinions and reactions. This is not a right-or-wrong issue. Since I am interested in what people think about computer literacy, all responses are valid! Reply to me directly if you don't think your message is appropriate for RISKS. (For instance, for my purposes, it is fine for lots of people to send messages just saying they agree with what someone else said, but such messages are best sent directly to me.) As usual, PGN [Peter G. Neumann] will publish in RISKS the most relevant submissions. In this case, he will also forward to me the other submissions on this topic.

All submissions are confidential. Anything that I quote or paraphrase will be presented anonymously, unless I get explicit permission from an individual to use his or her name. Usually I don't attribute a comment more specifically than to say, for instance, it is from "a Computer Science professor." You can indicate in your message the sort of work you do with computers, if you like.

In a 1985 school survey, 96% of the respondents - classroom teachers, computer coordinators, and administrators - said that their schools offered instruction in computer literacy. What do you know about course content and materials, school hardware and software, teacher training, and so on? Are your children learning about computers in schools? Have you been involved in any sort of school advisory committee for computer

education? If computer-literacy education has not crossed your path, what do you guess is taught in a typical class?

RISKS-LIST: RISKS-FORUM Digest Saturday 12 November 1988 Volume 7:Issue 76

Date: Wed, 9 Nov 88 16:43:58 EST

From: ronni@juicy-juice.lcs.mit.edu (Ronni Rosenberg)

Subject: Computer Literacy #2

A typical computer-literacy course has the following components. What do you think of this operational description of computer literacy? Should other topics be taught instead? in addition to these? Should everyone study this? As before, send mail to RISKS or to me, depending on your preference.

1. TERMINOLOGY AND JARGON: This is designed to enable students to "talk knowledgeably" about computers. Here are some definitions of a computer, from computer-literacy textbooks:

- "A computer is an electronic machine that solves problems or answers questions."
- "A computer ... is a machine that can handle large amounts of information and work with amazing speed."
- "A computer is an electronic tool that helps people do many different things faster, easier, or better."

Here is another definition, from a graduate computer-literacy class (for teachers, administrators, computer coordinators, and so on):

- "An operating system is a program that tells the computer how to deal with information - tells it how to move information, how to operate, how to do things. ... An operating system is done in a lower level language, machine language. It really controls the flow of electricity through the circuits."

2. **HARDWARE:** This is designed to give students a “working knowledge of computer equipment.” Typical classes use Apple IIs. The last large surveys show a national average of 1 machine per 40 students (grades K-12). Many schools cannot afford two disk drives per machine. A computer lab of 10-30 machines might have 1-3 printers (dot matrix). Devices such as joysticks, mice, and touch-sensitive displays are too expensive for most schools to buy (or these devices operate on machines that are too expensive), but some schools buy one of each, to pass around a class. The emphasis is on identifying components and handling equipment (e.g., floppy disks). A 1988 survey showed that among 11th grade students:

- 30% did not know what a cursor does,
- 60% did not know what a modem does, and
- 40% could not identify a spreadsheet as a software component or a video display as a hardware component.

3. **SOFTWARE:** Exposure to “basic software concepts” is designed to enable students to use computers as tools and to “remove the mystery of computers.” Typical classes show programs for word processing (which predominates), spreadsheets, and databases – ones that run on Apple IIs (or, in many cases, smaller machines). Students do not see actual program documentation. The emphasis is on the syntax of program commands. The survey cited above showed that students scored

- 72% correct on word-processing questions
- 52% correct on spreadsheet questions
- 31% correct on database questions

4. **PROGRAMMING:** When included, this means BASIC or LOGO programming. Again, the emphasis is on learning the syntax of some language commands. Miniscule programs (e.g., 10 lines) predominate.

5. JOBS IN COMPUTING: When included, this means brief discussion of careers in computing. There is a widespread sense that "computer literacy" is the passport to well paying jobs.

6. SOCIAL IMPACTS OF COMPUTERS: When included, this encompasses computer uses, ethics, and legal implications. Under uses, students might be told, for instance, that the FBI uses computers to store data on criminals and crimes, but not about privacy risks, data quality problems, etc. Students might be told that some people lose jobs because of computer automation -- and that these people can get other jobs working with computers. Ethics means a mention of computer crime. Legal implications means a warning that students should not copy software disks they use in class. Students are explicitly encouraged to have "positive attitudes about computers." Topics not covered include whistleblowing, the military influence on the computer profession, limits of simulations, and risks of large computer systems.

RISKS-LIST: RISKS-FORUM Digest Thursday 3 November 1988 Volume 7 : Issue 83

Date: Mon, 28 Nov 88 12:36:39 EST

From: ronni@juicy-juice.lcs.mit.edu (Ronni Rosenberg)

Subject: Computer Literacy #3

Expenditures of time and money on computer-literacy education represent important tradeoffs for schools. If you think that computer literacy should be taught in school, how do you think schools should pay for it (hardware, software, training, maintenance)? How should computer-literacy courses be fit into the school day?

Since school budgets and days are finite, these questions raise the issue of priorities. Should computer-literacy education be a high priority for our education system? Why or why not? How do you compare computer literacy with current education priorities?

RISKS-LIST: RISKS-FORUM Digest Tuesday 6 December 1988 Volume 7 : Issue 89

Date: Tue, 6 Dec 88 15:12:11 EST

From: ronni@wheaties.ai.mit.edu (Ronni Rosenberg)

Subject: Computer Literacy #4

What are your reactions to a proposal for a different sort of "computer literacy" course, described below? (I am not saying that all schools should teach such a course.) Is it a good or bad idea? Why? Should the description be changed? If so, how? How do you compare this with what you know about existing computer-literacy courses? Who should develop such a curriculum? Who should pay for it? Please respond directly to me. Thanks.

Compared to current computer-literacy classes, the proposed course would spend much less time reviewing the mechanics of operating machines, the syntax of applications software or programming languages, and rote learning of lists, from computer components to uses. It would spend much more time considering the capabilities and limitations of computers, through discussions of the impacts of important computer applications. This might be a standalone course or a series of discussions interwoven into courses in, for instance, social studies or history.

One specific example of material that could contribute to meaningful education about computers is a multi-media presentation entitled "Reliability and Risk: Computers and Nuclear War," produced and distributed by CPSR. The presentation explains how current political and military trends decrease the time allowed for people to react to a crisis, thereby shifting critical decision-making responsibilities to computers. It attacks the myth of computer infallibility by describing different types of computer errors, their sources and consequences. It explores the growing reliance on computerized decision-making and how a computer error, especially in times of crisis, could trigger an accidental nuclear war. Lasting a half-hour, the program obviously cannot cover the topic in great depth. But it does present salient points about an important and complex area of computer use, greatly heightens people's awareness of problems that they are unlikely to

learn about from magazine articles about computers, and stimulates exciting discussions and further thought. The presentation uses no computers, and the intended audience need no previous computer experience.

The proposed course might include discussions of

- SDI's computing requirements – so students could consider the concept of software trustworthiness and the potential for design errors in complex systems.
- The Vincennes episode – so they could consider the difficulty of using a system outside the boundaries of its intended use.
- The FBI's National Crime Information Center (NCIC) – so they could consider the relationship between civil liberties and computer technology.
- The National Test Bed, war games – so they could consider the limits of computer simulations.
- Computerized monitoring techniques – so they could consider impacts of computers on the workplace.
- How computer science is funded – so they could consider which sorts of problems society views as important.
- Some of the myriad of RISKS stories – so they could consider the risks of depending on computer systems.

And so on. Overall, the course would emphasize the importance of the social and political context in which a computer system is developed and used.

Appendix C

Curriculum Vitae

Ronni Rosenberg

Education

- 2/89 Ph.D., Science, Technology, and Society,
Electrical Engineering and Computer Science Dept., MIT
Thesis: "Computer Literacy Education"
- 2/80 S.M., Electrical Engineering and Computer Science, MIT
Thesis: "Incomprehensible Computer Systems: Knowledge Without
Wisdom"
- 5/77 B.A., Honors Program, Villanova University

Experience

- 1/88-present **Xerox Corporation,**
Advanced Information Technology Division
Consultant, Technical and Marketing Communications

Continue much of the work described below, since Xerox acquired the
Advanced Information Technology Division of Computer Corporation
of America. Also advise on procedures for producing and organizing
technical and marketing materials.
- 2/80-12/87 **Computer Corporation of America,**
Advanced Information Technology Division
Manager of Communications and Computer Scientist

Managed Advanced Information Technology publications, technical
writing, text processing, and library functions. Had primary respon-
sibility for technical writing and documentation design in this R&D

group, including journal articles, conference papers, user manuals, project documentation, CCA Technical Reports, project overviews, and marketing collateral. Managed proposal efforts ranging up to multi-million-dollar proposals involving CCA and several other companies. Developed proposal preparation materials – procedural checklist for proposal managers and proposal template – used by entire Division. Designed and wrote several tutorial manuals (Unix, Emacs, and Troff/Nroff). Designed, wrote, and produced videotapes for technical and marketing purposes. Contributed technically to selected research projects, including CAD/CAM Database Management System, Social Security Administration Database Architecture Study, and NBS Query Language and Database Architecture Project.

9/78-1/80 **MIT, Laboratory for Computer Science**
Research Assistant

Supervised by Professor Joseph Weizenbaum. First MIT Computer Science student to receive an advanced degree in Societal Implications of Computers.

9/78-5/78 **MIT, Program in Science, Technology, and Society**
Research Assistant

Involved in NSF-funded project on "The Impact of the Computer Presence on the Individual" (principal investigator: Professor Sherry Turkle). Contributed to the initial phase of the project and interviewed computer hobbyists.

9/77-1/78 **MIT, Department of Electrical Engineering and Computer Science**
Teaching Assistant

Taught sections of introductory course for Computer Science undergraduates.

9/75-5/77 **Villanova University, Department of Mathematics**
Computer Laboratory Assistant

Summer 1976 **Villanova University, Honors Program**
Research Assistant to the Director

1/75-9/75 Mathematics Tutor (self-employed)

Awards

1987-1988 Alfred Kiel Fellowship for the Wiser Uses of Science and Technology

1986-1987 GTE Graduate Fellowship

Publications: Societal Impacts of Computers

Ronni Rosenberg, "Computer Literacy Education," *Bulletin of Science, Technology, and Society*, 7, 5 & 6 (1988, mistakenly printed 1987) 984-994.

Ronni Rosenberg, "Computer Literacy in Schools," Symposium on Directions and Implications of Advanced Computing (DIAC-88), St. Paul, Minnesota: 21 August 1988.

Ronni Rosenberg, "Computer Literacy: A Study of Primary and Secondary Schools," Third National Conference on Technological Literacy, Arlington, Virginia: 5-7 February 1988.

Ronni Rosenberg, "A Critical Analysis of Research on Intelligent Tutoring Systems," *Educational Technology* (November 1987).

Ronni Rosenberg, "Privacy in the Computer Age," *The CPSR Newsletter*, 4, 3 (Summer 1986), 4, 4 (Fall 1986), and 5, 1 (Winter 1987). Computer Professionals for Social Responsibility.

Publications: Computer Science

Sunil Sarin, Mark DeWitt, and Ronni Rosenberg, "Overview of SHARD: A System for Highly Available Replicated Data," Technical Report 162, Xerox Advanced Information Technology (May 1988).

Umeshwar Dayal, et al., "Overview of PROBE: An Object-Oriented, Extensible Database System," *IEEE Expert*, invited paper for special issue on expert systems (March 1988).

Ronni Rosenberg and David Reiner, "Database Design and Evaluation Workbench (DDEW)," 3/4" videotape, 16:14, Xerox Advanced Information Technology (29 October 1986).

Sy Danberg, Stephen Fox, Ronni L. Rosenberg, and Mark Vincenzes, "LDM Operations Manual for VAX/VMS," Technical Report XAIT-84-10, Xerox Advanced Information Technology (November 1984).

David Goldhirsch, Terry Landers, Ronni L. Rosenberg, and Laura Yedwab, "MULTI-BASE System Administrator's Guide," Technical Report XAIT-84-09, Xerox Advanced Information Technology (November 1984).

Terry Landers, Stephen A. Fox, Daniel R. Ries, and Ronni L. Rosenberg, "Daplex User's Manual," Technical Report XAIT-84-01, Xerox Advanced Information Technology (March 1984).

Ronni Rosenberg, "U.S.S. Carl Vinson Spatial Data Management System," 3/4" videotape, 9:46 (13 September 1983).

Terry Landers and Ronni L. Rosenberg, "An Overview of MULTIBASE," in H. J. Schneider, ed., *Proceedings 2nd International Symposium on Distributed Databases*, Berlin, West Germany, 1-3 September 1982 (XAIT Reference Number 91).

Christopher F. Herot, Richard Carling, Mark Friedell, David Kramlich, and Ronni L. Rosenberg, "Overview of the Spatial Data Management System," Technical Report XAIT-81-08, Xerox Advanced Information Technology (November 1981).

Laura Yedwab, Christopher F. Herot, Ronni L. Rosenberg, and Carol Gross, "The Automated Desk," *Proceedings of the ACM SIGMOD/SIGSMALL Workshop for Databases for Small Systems*, Orlando, Florida, 14-15 October 1981. Reprinted in **Tutorial: End User Facilities in the 1980s**, James A. Larson, ed. (California: IEEE Computer Society, 1982).

Richard Winter, Tomas Lozano-Perez, Donald E. Eastlake III, and Ronni L. Rosenberg, "Overview of the Chemical Substances Information Network," Technical Report XAIT-81-03, Xerox Advanced Information Technology (1 April 1981).

Ronni Rosenberg, "CSIN: Chemical Substances Information Network," 1/2" videotape, 13:00 (18 May 1981).

OFFICIAL DISTRIBUTION LIST

Director 2 copies
Information Processing Techniques Office
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Office of Naval Research 2 copies
800 North Quincy Street
Arlington, VA 22217
Attn: Dr. R. Grafton, Code 433

Director, Code 2627 6 copies
Naval Research Laboratory
Washington, DC 20375

Defense Technical Information Center 12 copies
Cameron Station
Alexandria, VA 22314

National Science Foundation 2 copies
Office of Computing Activities
1800 G. Street, N.W.
Washington, DC 20550
Attn: Program Director

Dr. E.B. Royce, Code 38 1 copy
Head, Research Department
Naval Weapons Center
China Lake, CA 93555