

Preparing Children for a Computer-Rich World

Computer-literate students will be able to distinguish between the routine tasks from which computers can release them and more sophisticated capabilities—framing and solving problems, communicating visually and verbally—that are least likely to be automated.

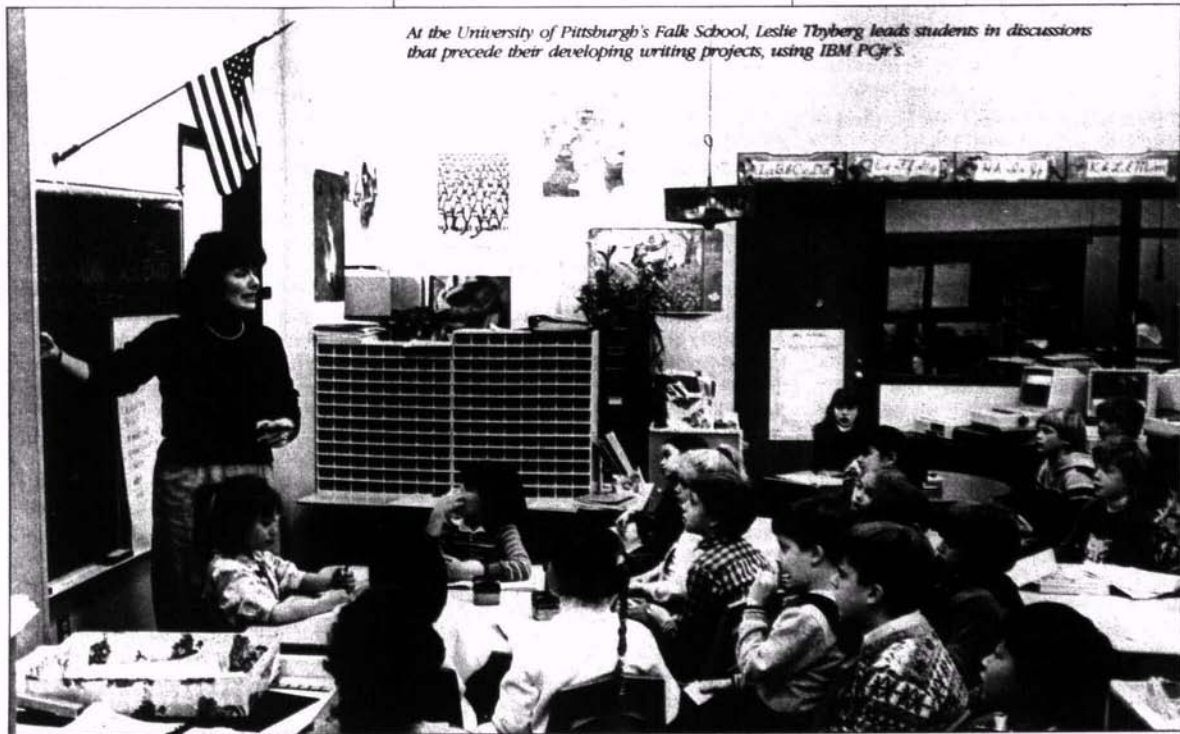
Which computer should we buy? Which software should we buy? How can we incorporate the computer into the curriculum? These questions, and many more detailed questions about different brands of computers, printers, hard disks, and other pieces of hardware, keep arising whenever I talk to people who know that I am an instructional

psychologist who also is heavily involved with computers. The same basic questions pop up whether I am talking to a principal or school superintendent planning for next year, a teacher wanting to be more creative and productive, a ministry of education official planning to introduce computers into the schools of an entire country, or my brother, who is

worried about whether his daughter will be a full member of our increasingly computerized society. When I try to answer such questions, I think about three things:

- how massive use of computers will change the occupational roles available in our society,
- what that implies in terms of changes in the curriculum, and

At the University of Pittsburgh's Falk School, Leslie Thyberg leads students in discussions that precede their developing writing projects, using IBM PCjr's.



“ ‘Computer literacy’ has little if anything to do with handling computers. It is, rather, a set of broad cognitive capabilities that allow one to think deeply, creatively, and efficiently, and to communicate the results of that thinking.”

● how the computer can be used to help bring about such changes.

How the Computer Is Changing Society

We have had major technological changes before. The invention of movable type and the steam engine brought changes as profound as those the computer will bring. Their long-term impact was to change the relative value of different skills. The key to surviving the transition from handwritten to printed books was not having the ability to read printed rather than handwritten words. Nor was the key to doing well in the Age of Steam the result of having played with a steam engine. Rather, doing well required having the skills to use the new technology to extend one's capability. To take advantage of cheaper books, one needed to be able to read with greater facility, since a book was no longer a rare object worthy of days of contemplation. To use steam engines, one had

to understand mechanics, but it was much safer and easier to acquire that understanding from experimentation other than playing with steam engines.

Similarly, we can be pretty sure that certain skills will become more important in a computer-rich, automated society and others, less valuable. So far, the pattern has been to replace routine mental work with computers, enabling people to engage in more creative intellectual activity, including design and problem solving in many technical areas. An indirect effect is that automation is displacing more of the labor force from manufacturing jobs and moving them into service jobs. These jobs require different social skills and generally more communication skills than work on an assembly line requires.

Implications for New Curriculum

A cruel but useful caricature is that of the role the assembly line assigned to schools. Schools were supposed to teach children to be on time, work according to a fixed schedule, obey orders, read and follow simple directions, and do simple arithmetic. In contrast, the computer era requires more complex schooling—good verbal and graphic communication skills and the ability to formally characterize and solve problems. This means that written composition skills are more important, along with the hard part of math and science, understanding and solving problems. While specific technical knowledge is needed in a computerized society, the specific facts and skills acquired today will be less important tomorrow. Thus, the curriculum must teach children to learn new information and skills efficiently. We can be pretty sure that the world of the future will be dominated by the person who is a “quick study.”

The school day is already used up; some think it is too short for the existing curriculum, and it seems inappropriate to add new items to it. We do not want to exchange 20 minutes of reading for 20 minutes of learning skills. Instead, we must teach, in the context of existing subjects, learning and problem-solving skills, the skills of communication, and certain social skills. In a sense, “computer literacy” has little if anything to do with handling computers. It is, rather, a set of

broad cognitive capabilities that allow one to think deeply, creatively, and efficiently and to communicate the results of that thinking.

Tools That Extend Thinking

Computers have two important roles in the classroom. First, they can be useful in teaching some of the skills needed for the computer era. They can provide instructional microworlds and minilaboratories within which students can carry out experiments. Second, they can help students tackle problems that are too big to be handled within the limits of human thinking. The best is yet to come in these areas, as artificial intelligence techniques start to be used, but in this article I am focusing on what is possible now.

Because schools use so many different kinds of computers and have so little money for software, few good programs are tailored to school needs. We can, however, consider the potential educational uses of software developed for the business market. The most pervasive use of personal computers, for example, is probably for word processing. A second major use is with spreadsheet programs, which help people explore alternative mathematical models easily. A third common use is probably with graph and chart programs that facilitate communication. In contrast to educational software, which is in its infancy, word processing, spreadsheet, and graphics programs are well developed, readily available, and cost-effective. All three types of software can be used in the schoolroom.

Word processing. Although we would like our children to be able to write better, many obstacles stand in the way of teaching effective writing. Teachers have too little time to read the scrawls of many students, and writing is so physically slow and tedious for some students that they are not able to generate texts substantial enough to build good writing skills. For the same reason, they are unlikely to revise what they have written, or even to edit for typographical and syntactic errors. The total effort involved when a child writes a composition and a teacher reads it is so high that facile interaction between teacher and student is difficult. The teacher's comments both on editing and content



"The artificial intelligence tools we are developing . . . will coach students in systematic learning skills that they can practice in microworld environments."

problems can swamp the student with too much simultaneous feedback. Having put forth so much effort to produce the composition, the student has little tolerance for correction.

Ready access to word processing, however, could give students much more practice in a skill that requires considerable practice. Teachers could read essays more quickly and thus be more productive. They could also respond to one or two issues at a time, and the student could then deal with the identified problem and print another copy of the text. Standard programs that highlight differences between the first and second computer file could be used to point out the student's changes and enable the teacher to reread the student's paper more quickly.

Even so, teaching writing skills remains a labor-intensive proposition. But tools now being developed can ease the teacher's problems considerably. One such tool is the spelling corrector, common in the business world. Students could use the spelling corrector as a tool, so that by the time

they turned in a paper, it would have no spelling errors or typos. The teacher who wants students to become better at spelling, however, may prefer that spelling correction not be automatic.

It is possible to write software that simply points to lines in which spelling errors have been found, leaving part of the detection and editing to the student. We have developed such a capability (written in C for a VAX computer) for use in University of Pittsburgh remedial writing courses. Initially, it directs the student to two or three lines within which a spelling error has been detected. If the student cannot find the error, the program points to the specific line in which it is located. If necessary, it can highlight the misspelled word.

It is possible to go one step further and detect syntactical errors also. For this, schools need special software different from that of the business world. Employers assume that workers understand English grammar and can correct an error once it is pointed out. For many school children who do not

understand all of English grammar, pointing out an error is not enough. They need special software that describes syntactical errors in language they can understand.

For example, a common error in student writing is the comma splice. In it, two independent clauses are connected with only a comma: *I got to school early, nobody was there.* Simply pointing out comma splice errors may not improve a student's performance, if the student does not seem to be forgetting to use a conjunction. When students are asked why they write sentences such as the one above, they usually answer that they were told that a sentence expresses a complete idea. They keep writing until they have put down the entire idea they had in mind. The comma may be put down because of another incorrect rule (or bug): "Insert a comma at those points in the sentence where you would pause if reading it." The most helpful feedback to the student would be to isolate the two clauses, point out that each of them is a complete idea, and talk about strategies for combining them grammatically. Simply saying that the *and* is missing may not help much.

This kind of responsiveness requires a form of syntactic analysis very different from grammar correction for the business world. The program must be smart enough to have a model of how a student generates such a sentence in the first place. This capability is becoming possible for the most common errors in remedial writing classes, but a complete system that can deal with all syntactical errors will not be ready for some time.

The hardest part of writing is getting a good set of ideas in the first place. Products (e.g., *Thinktank*) that facilitate outline construction are already available. These can be very useful. Some people might object that an outline can be written on a piece of paper as easily as on a computer. This is true only if you already know what you want to write about. If you are still formulating ideas, you may need to insert new entries between the existing ones and move whole sections from one part of the outline to another. The special advantages of an outline processor are automatic numbering of entries, automatic indenting, and the ability to reprint the outline with or without different levels of de-

tail. While these are all useful features, they are not essential. Children can use existing word processing equipment to develop outlines and plan essays.

As a college professor, I read quite a few student papers. I am struck by the number of students who have mastered spelling and syntax but who have not yet learned how to develop an argument clearly. It could be very useful to give students exercises in developing ideas for an essay using word processing software as a tool. Since some of my students have computers at home, I give them suggestions for developing essay assignments, such as those in Table 1. These can work quite well regardless of their word processing program. Simple text editors, *Bank Street Writer*, *WORD*, and *Wordstar* can all be used.

Of course, even better idea-development software will be available in the future. Researchers at the Xerox Palo Alto Research Center have already developed a product called *Notecards* that allows both word processing and graphics to be used on many small pieces of text, computer-screen equivalents of the note cards we sometimes use to prepare an article. Further, it can indicate relationships between part of one note and another separate note. For example, one could indicate the evidence for an assertion by showing a relationship between that assertion on one note card and the evidence for it on another. One could even show relationships between text and graphics. For example, one can have a map of Florida on one card and show a relationship between a description of the Okefenokee Swamp on another card and the location of the swamp on the map. *Notecards* will even turn a set of notes into a first draft. Unfortunately, it will be a few years before most schools can afford the more powerful computer needed to run *Notecards*.

Spreadsheet and graphing programs. The spreadsheet is a system that allows data to be represented as a matrix. Various operations can be performed on selected columns of the data matrix to yield new figures. For example, if each row of the matrix represents sales figures for one month and the columns represent gross receipts and costs, the costs column can be subtracted from the receipts col-

umn to produce profit figures. Spreadsheet programs can compute summary statistics on columns, make correlations between columns, and graphically display single columns and intercolumn relations.

A spreadsheet program can be very useful in science courses in which children conduct experiments, manipulating certain parameters to produce measurable outcomes. If the data are systematically kept in matrix form using a spreadsheet program, the student can examine and manipulate them. Consider a simple balance beam experiment. Suppose the student hangs different weights at different distances from the fulcrum and records all relevant data from every arrangement that balances. This means that each balanced situation will be represented by a row in the matrix, and the columns will correspond to the mass of the left weight, its distance from the fulcrum, the mass of the right weight, and its distance from the fulcrum. Once these figures are accumulated, the teacher can guide the student through some experimentation with the spreadsheet program. Do the two mass columns correlate? Do they produce an interesting scatterplot? What about the distance measures? What about comparing mass plus distance on the left with mass plus distance on the right? If there is still no clear pattern, you may want to add some of your own data to that gathered by the student. Finally, the student can be asked to compare the product of distance times mass on the left with the product of distance times mass from the right. This shows a clear pattern that makes sense when plotted.

In our own work at the Learning Research and Development Center, we are adding spreadsheet capability to a variety of exploratory microworlds, computer programs that simulate experiments such as that just described with the balance beam. For example, Valerie Shute has developed an economics microworld in which the student can vary income, preferences, price, and other parameters to project the effects on supply and demand. In Peter Reimann's optics world, the student can vary refraction index and lens curvature to observe their effect on a refracted light beam. The spreadsheets in these microworlds supply a "lab notebook" that supports systematic experimentation. The simulation microworlds themselves can probably be implemented for the larger computers already found in schools. The artificial intelligence tools we are developing (which require more powerful computers that schools won't have for another few years) will coach students in systematic learning skills that they can practice in microworld environments.

Programming. The simple, low-level programming courses taught in schools today will not help students secure places in the work force. In the future, routine programming will be done automatically by the computers themselves. Already, several large corporations are eliminating their data processing departments in favor of massive database systems that can respond to questions in a form comparable to standard English. Consequently, if programming is to be taught in our schools, it must be for purposes other than vocational training.

Table 1: Suggestions for Preparing Written Assignments

- Start with an outline in which you list your main points.
- Under each main point, list arguments for and against that point.
- Evaluate the arguments for and against each main point and make a decision about whether to modify your point.
- Restate your evidence for and against each point to take account of any modifications.
- Reorder your points taking account of the evidence for and against each. Put points that rely on related evidence close together. Put points in a coherent order.
- Write the center of your essay, in which you present your assertions, present evidence for and against them, and critique them in light of the evidence.
- Write a conclusion that summarizes the supported thread of your argument.
- Write an introduction that states the issue being discussed and the kinds of points you make in the main part of your essay.

Using the Computer to Prepare Future Teachers

In Ed 302, Curriculum and Instruction at Allentown College of St. Francis de Sales, future teachers receive pragmatic instruction in using computers as an object of instruction, a manager of instruction, and an aid to instruction. The course, which focuses on lesson planning, teaching methodology, and micro-teaching, is designed to help future teachers learn in relatively short time how to use computers to produce highly useful teaching tools.

The Computer as Object of Instruction

Future teachers are introduced to the three most popular brands of computers being used in schools, the TRS-80, Apple, and IBM PC or PC compatible. Their first goal is to accomplish an ordinary task, the recording of grades and generation of report cards. Students are asked to create and print out a list of ten student names, ten grades, and an average for each. They receive a clear set of written instructions on how to perform each required step, beginning with turning the computer on and putting in disks. During the class period, each student sits at a separate terminal and follows along with the instructor, who explains exactly what to do.

The Computer as Manager of Instruction

In this section, students are introduced to *VisiCalc*, the electronic spread sheet, which enables them to create a rollbook. They label columns on the spread sheet for recording grades for tests, class participation, and so on. Once the grades have been typed in, averages can be automatically calculated for each class member. The students' homework assignment is to give a relative weight for each grade, for instance, quizzes equal 20 percent of the final grade, essays 20 percent, tests 30 percent, and so on. The formula is edited to include separate multipliers for each column and can be used again by simply changing names and grades.

The Computer as Aid to Instruction

In this segment of the course, the future teachers learn how to generate written material using IBM's word processing package, *Writing Assistant*. With this software, students learn to center; check spelling; search, replace, and move paragraphs. Again, they are given concise written instructions for completing each step. Their home assignment is to produce a one-page, multiple choice, short essay quiz that is then typed on a ditto master using a conventional typewriter and reproduced for each class member. Considering the difficulty of producing a letter-perfect ditto master, especially if one is not an ace typist, the object of this assignment is for students to see that a computer and printer are excellent aids for correcting and re-running ditto masters.

Students are also introduced to an authoring system using Radio Shack's *Quick Quiz*, which provides programmed test formats. Once all test questions and correct answers have been typed in, *Quick Quiz* produces an answer key.

As an aid in their own teaching training, the students are required to run tests on the effectiveness of selected teaching strategies, which they try out on their college peers in 15-minute teaching sessions. They first identify a few specific behavioral objectives to be addressed and prepare pretests and posttests using items criterion-referenced to these objectives. Following their teaching session, the test scores are tallied for each of the future teachers' "students," and grades are calculated for each objective. A hand analysis is then done to find significant differences in pre- and posttest scores for each objective. Finally, they again use *VisiCalc* to perform the same analysis on the computer. Thus, these student teachers learn not only how to do a statistical analysis of their own performance, but they also learn firsthand the benefits of using a computer to carry out the same task.

—By Andrew Robert McGilvray, assistant professor and chair, Department of Education, Allentown College of St. Francis de Sales, Center Valley, PA 18034.

The most important reason to be able to program is to put the computer to work for oneself. Thus, the products of programming exercises should probably be software that students will use later as tools in another course. Students use this approach at Stevens Institute of Technology. In their first year at Stevens students accumulate a tool kit of programs that they use in engineering courses, science courses, and advanced mathematics courses. By structuring the curriculum this way, Stevens has ensured that students learn not only how to program, but also when to program and why. Further, when they have to use the software they wrote a year earlier, students learn that, rather than just being a stodgy requirement of the instructor, good programming style benefits them.

The same approach could readily be taken in high school, with teams of students writing graphing and plotting programs, sorting programs, and data accumulation programs for use in later courses. As teachers become more facile with computers, they may use this approach outside mathematics and science to enable students to create outline development programs, word search programs, and programs to help them study foreign language vocabulary. What is critical is not the exact program content but its usefulness to the student, providing a better rationale for computer use in the first place.

Developing Higher-Level Thinking and Learning

Standard computer software can help children become better at acquiring knowledge, solving problems, and communicating their ideas. Computers should be used as assistants to facilitate and extend learning and problem solving. When used this way, computers will certainly help children prepare for a high technology future by assisting them to develop those higher-level thinking and learning processes that are least likely to be automated into obsolescence and most likely to benefit from computers that extend human capabilities. □

Alan M. Lesgold is professor of psychology and senior scientist, Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA 15260.

Copyright © 1986 by the Association for Supervision and Curriculum Development. All rights reserved.