DOCUMENT RESUME

ED 268 013 SE 046 555

AUTHOR	Thompson, Charles, Ed.; Vaughan, Larry, Ed.
TITLE	Computers in the Classroom: Experiences Teaching with Flexible Tools. Teachers Writing to Teachers Series.
INSTITUTION	Northeast Regional Exchange, Inc., Chelmsford, MA.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC.
PUB DATE	Jan 86
GRANT	NIE-G-82-0017
NOTE	164p.
PUB TYPE	Books (010) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC07 Plus Postage.
DESCRIPTORS	Computer Oriented Programs; *Computer Software; *Databases; Elementary Secondary Education; Problem Solving; *Programing; Science Education; Social Studies; *Word Processing; *Writing Instruction
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

First-hand accounts of what teachers have done with students and computers in their classrooms, how students have responded, and what and how teachers have learned from these experiences are discussed in the 19 articles in this book. The articles are presented under these headings: (1) teaching writing with word processors; (2) learning to inquire through database programs (considering integrating database programs across the curriculum databases throughout and beyond a junior high school, and IMMIGRANT-a social studies simulation for AppleWorks); (3) other applications that make room for teachers (addressing microcomputer-based laboratories, filling a gap in reading comprehension programs, and musical computing); and (4) learning to program and programming to learn (discussing 3 years with Logo in a third-grade classroom, mathematics classroom problem-solving with notebook computers, enriching mathematics with computer activities, selecting a writing adventure, and other areas). Summaries of the articles are provided in an introduction. (JN)

Computers in the Classroom:
Experiences Teaching with Flexible Tools

Teachers Writing to Teachers Series

Edited by Charles Thompson
and Larry Vaughan

Northeast Regional Exchange, Inc.

The National Institute of Education
U.S. Department of Education
Washington, D.C. 20208

Computers in the Classroom:
Experiences Teaching with Flexible Tools

Teachers Writing to Teachers Series

Edited by Charles Thompson
and Larry Vaughan
Assistant editor Elsa Martz

Northeast Regional Exchange, Inc.

The National Institute of Education
U.S. Department of Education
Washington, D.C.

Northeast Regional Exchange, Inc.

The Northeast Regional Exchange, Inc. (NEREX), a private, not-for-profit corporation, is a service and research agency that promotes educational equity and improvement. NEREX coordinates resources and sharing of information among the seven state departments of education in the Northeast based on an established set of state and regional priorities. Through NEREX, states are able to expand their available resource base and work through regional sharing efforts toward program improvement in local school districts and other educational institutions. The Northeast Regional Exchange, Inc. is governed by a Board of Directors that includes the seven Chief State School Officers from the Northeast and eight representatives from a wide variety of education constituency groups in the region.

Board of Directors

The development of this publication was supported by a grant from the National Institute of Education, U.S. Department of Education (NIE G-82-0017). The opinions expressed in this publication do not necessarily reflect the position or policies of the U.S. Department of Education or the Northeast Regional Exchange, Inc. Dissemination of this publication was approved by the U.S. Department of Education's Publication and Audiovisual Advisory Council (PAVAC).

CONTENTS

INTRODUCTION

Stories to think with, Seymour Papert and our contributor Molly Watt might call them, these articles by teachers about their experiences using computers in their classrooms. Not instructions by experts about how to do it. Not theories or research reports, though we have nothing against theory and plead guilty to committing research from time to time, ourselves. Not even software reviews. First-hand accounts of what teachers have done with kids and computers in the classroom, how kids responded, and what and how teachers have learned from the experiences.

In other words, "Teachers Writing to Teachers"—which incidentally is the name of a series of books produced by NEREX as resources for aspiring and practicing teachers. The first book in the series was **Understanding Writing: Ways of Observing, Learning and Teaching.** After reprinting this volume several times, we at NEREX were convinced that teachers did, indeed, value learning about the experiences of their peers. Direct feedback told us that more such craft knowledge was needed in the field. The second in the series was **Teaching Mathematics: Strategies that Work;** and now there are others. Each comprises several articles written **by** teachers **for** teachers. In each, the chapters cover a spectrum of grade levels and topics. Each collection is bound together by its focus on classroom life and teacher reflections about how learning takes place.

The articles in this volume include some examples of promising practices, but from our point of view, they are offered more to illustrate interesting approaches than to encourage replication of the details of the practices, themselves. They are stories with which or in terms of which other teachers may think about potential uses of computers in their own classrooms. The articles also illustrate the processes through which teachers develop and adapt new practices as well as highlighting the discoveries that they make along the way. So the stories can also be used for inspiration and guidance about the processes of development, adaptation, and discovery in the classroom.

So, stories to think with.

The central characters in virtually all of these stories are teachers and kids, not computers. The computers are certainly important—they are the sets and props or maybe even the stage on which the stories are played out. Frequently the computer is also the Muse who inspired the writer. But computers are not the protagonists. Teachers are.

Teachers may be the protagonists of these stories because we prefer to hear about and promote uses of the computer as a tool for teachers and students, rather than uses in which teachers and students become tools of the computer. There are times when one may want to surrender to the computer control over the pace and path of instruction, and some such times occur in these articles. In general, however, computers and software play supporting roles here, opening doors and performing the drudge work so that teachers and students can think, learn, and make things.

By playing a supporting role, computers frequently transform the relationships among the central characters. Teachers end up learning along with—even **from**—their students. "Handicapped" or "special needs" students sometimes teach their "normal" peers as well as their teachers. Teachers lose control over students, not to chaos but to absorption in learning. And a variety of other shifts, anticipated and unanticipated, subtle and unsubtle, are precipitated by the arrival of the machines.

Perhaps that's not quite right. The computers don't really transform most of these relationships. It's teachers and students who see and seize the opportunities afforded by the machines that really do the transforming. In virtually every article published here, one can imagine ways in which inflexibility or lack of imagination might have stalled out the changes. Fortunately for us, flexibility and plenty of imagination prevailed.

The teacher-writers represented here range from a self-proclaimed "computer illiterate" to trainers of teachers with several years of experience in the classroom, in teacher training courses and workshops, and in research and development projects at the cutting edge of thinking in the field. The breadth of range is deliberate. Nearly any teacher can find within the collection at least a few articles appropriate to his or her own level of experience and sophistication For our own part, we found articles at both extremes equally exhilarating, and we believe that those who put aside their fears and/or pride in expertise will be rewarded by the whole range, as well.

Although we chose the articles for their independent interest, placing emphasis on the degree to which they recounted telling experiences and offered insightful reflections on the experience, the collection fell readily into easily-grasped categories for us.

Teaching Writing with Word Processors

Sharon Vinciguerra's "Computer Iliterate Mades Good," and "Word Processing—Another View from Serendip" by Sherry Anderson and Jack Halapin both communicate the excitement of computer-inexperienced teachers learning along with their students. Vinciguerra's high school special needs students published a newspaper and used a print utility to produce a profusion of print products that drew curious visitors from all over their school. Anderson and Halapin started from scratch and still beat the four-week deadline for their middle schoolers' magazine. In "Electronic Newspaper: Online with Thurston," Roxanne Baxter Mendrinos tells how students in her junior high found an audience in two other Massachusetts schools—and found a reason for writing in the bargain.

Carolyn Kohli's "Composing with the Word Processor" and Michael O. Stegman's "The Craft of Processed Words" both deal in sophisticated ways with the teaching of writing at the high school level. Kohli describes her classroom as a writing workshop, emphasizing students' critiquing each others' work in pairs and small groups. Stegman describes a variety of very intriguing exercises designed to help students learn to make good use of the word processor's essential functions as they write and rewrite.

Learning to Inquire Through Database Programs

Database management programs—programs that permit the user to organize, store, search, and retrieve information—offer teachers and students a range of teaching and learning opportunities as limitless as those offered by the word processor. Databases are to inquiry skills as word processors are to writing.

For those new to databases as well as to teaching with them, Howard Moskowitz ("Integrating Database Programs Across the Curriculum") provides a nice introduction, following through with examples of using databases at three progressively more sophisticated levels of teacher and student understanding across a number of subject areas. Roxanne Mendrinos' "Databases Throughout and Beyond a Junior High Classroom" shows how communication software can let students reach beyond the classroom to a world of current data.

In "IMMIGRANT," Don Morrison and Joseph Walters explain how they and several colleagues at the Harvard-based Educational Technology Center created a social studies unit that uses an integrated software package, including a spreadsheet and a word processor along with the database manager, to offer students a chance to follow an actual immigrant after arrival in Boston in the middle 1800's.

John Rouleau reports a very different use for a database program—as a tool for the teacher bent on transforming his classroom through a management system for instructional individualization. Here the computer facilitates implementation of a popular idea—individualization—that is far easier to espouse than to put into practice, and does so by supporting rather than supplanting the teacher.

Other applications That Make Room for Teachers

This section includes three articles that discuss [illegible] open-ended or tool programs in reading, science, and music. "Filling a Gap in Reading Comprehension

9

Programs" by Rebecca Corwin, Joanne Bronga, and Charlotte Reid reports on an emerging set of programs that go beyond drill and practice in reading instruction, programs that fill a major gap by leaving things out. The programs, in which the learner uses contextual clues to fill in the missing parts of selected texts, are examined to identify key features of good reading software.

In "Microcomputer-based Laboratories: Connecting Computers and Science," Ted Hall describes his experiences with software and peripherals that permit students to measure a broad array of phenomena—from heart rates to motion to pH—and manipulate the resulting data.

Fred Schouten's discussion of computers in the music classroom, "Musical Computing," includes several programs that are changing musical composition and composition instruction as word processors are changing writing and writing instruction. For certain types of performance skills and related concepts, Schouten also finds value in commercial drill and practice programs.

Learning to Program and Programming to Learn

Linda Colvin's "Three Years with Logo in a Third Grade Classroom" traces the evolution of her own thinking about and skill in teaching Logo, from novice to expert in the eyes of her peers. Molly Watt's "Putting the Fun Back into Recursion" offers a way of dramatizing one of the most sophisticated and elusive concepts in programming.

While Colvin and Watt address problems in teaching programming. the next three articles in this section—Dawes Potter's "Math Classroom Problem Solving with Notebook Computers," Eileen Backofen and Barbara Bussey Ringgold's "Enriching Mathematics with Computer Activities," and Theresa M. Reardon's "What Would Happen If . . .?"—describe experiences in using programming to teach concepts in mathematics. Perhaps the idea of teaching mathematics through programming will not surprise you, but how about teaching **writing** through programming? That's the subject of Anne Weber Dunham's "Choosing a Writing Adventure."

The final article in this section, John B. Howe's "Plagiarism: Will the Computer Be to Blame?" raises and suggests a practical, unemotional way of dealing with a troublesome issue in all kinds of activities that require students to demonstrate their competence in programming by creating original programs. Howe's article represents a particularly pointed example of a phenomenon occurring throughout this collection: the frequently unglamorous on-the-job problem solving necessary as teachers put computers to work in their classrooms.

One of the threads that helps to bind together all nineteen articles has to do with our subtitle, "Experiences Teaching with Flexible Tools." Though most computer-using teachers have just begun to integrate tool programs such as word processing, databases, graphics, graphing and spreadsheets into their instruction, we believe tool applications can be used creatively by teachers and students in ways that factor in additional motivation and excitement for learning. These tools let learners control the directions of their learning. With such tools students can manipulate their own ideas and organize their thinking. Tools can allow learners to reflect on the same phenomena from different perspectives, to try and retry without starting from scratch, and they can be used to extend knowledge acquisition far beyond the walls of the school.

We hope the experiences teaching with flexible tools described in the chapters that follow help to stimulate your own creative use of a variety of tools for learning.

Charles Thompson
Larry Vaughan

January 1986

About the Editors

Charles L. Thompson is Co-Director of the Educational Technology Center at the Harvard Graduate School of Education. ETC was created by the National Institute of Education in 1983 to research ways of using information technologies to improve education in science, mathematics, and computing at the elementary and secondary levels. The Center conducts research through collaboration among teachers, subject matter experts, social scientists, and people with expertise in information technologies. Most research teams are addressing "targets of difficulty" within the standard curriculum—topics that confuse and discourage many students and frustrate teachers' best instructional efforts, as well. ETC hopes by finding technology-based ways of breaking through perennial obstacles to enable many more students to persist and succeed in scientific and quantitative studies.

Before assuming the Co-Directorship of ETC, Dr. Thompson directed the learning technology unit at Education Development Center in Newton, Massachusetts. At EDC, he worked with Judah Schwartz to initiate a mathematics software series—including **SemCalc** and **The Geometric Supposer**—published by Sunburst Communications. Other EDC products initiated by Dr. Thompson include two programs developed by Ilene Kantrov and Jay McCloskey: **Writewell,** a tool for use in writing instruction, and **Kody's Jungle Adventure,** an early reading instruction program involving speech recognition, speech output, graphics, music, and text.

Earlier, at The NETWORK in Andover, Massachusetts, Dr. Thompson had managed a major study of federal dissemination programs designed to support school improvement. and before that he served for four years in the Educational Policy and Organization program at the National Institute of Education. Charles Thompson's principal research interests are in the process of educational change, including the introduction of computers and other information technologies into schools and classrooms.

Larry Vaughan is Manager for Research and Technical Assistance at the Northeast Regional Exchange, Inc., a non-profit educational research and service agency for a seven state region. Among his responsibilities is overseeing regional instructional technology improvement programs. He has been involved with computer instruction for over fifteen years, beginning in graduate school with evaluation studies of "The Talking Typewriter." At the University of Pittsburgh's Learning Research and Development Center, he worked extensively with the Experimental Time Sharing System conducting research on software applications for elementary school teachers and students.

During the past five years, Mr. Vaughan has concentrated on helping schools make effective use of instructional technology in curriculum areas. He has coordinated several regional computer instruction conferences, convened educational technology task forces, and produced and edited a number of resource books for educators including **Technology Programs That Work** and **Evaluation of Educational Software: A Guide to Guides.** In 1985 Mr. Vaughan was the executive producer of "Computers in Education: A Mid-80's View," a series of three 90-minute video programs broadcast via satellite to a nationwide audience. Also, he developed an electronic newsletter for educators on The SOURCE called "TECH TALK." Larry Vaughan's most recent work has focused on promoting the use of tool software within the existing curriculum.

I TEACHING WRITING WITH WORD PROCESSORS

COMPUTER ILLITERATE MAKES GOOD

by Sharon Vinciguerra

Editor's note: A single computer transformed the relationship between self-styled computer illiterate Sharon Vinciguerra and her high school special education students. Or rather, Vinciguerra's willingness to expose her initial ignorance and learn alongside her students made the transformation possible. Using a word processor and a utility program to create illustrative and decorative designs, Vinciguerra's "handicapped" students not only put out a newspaper, but also printed stationery, business cards, birthday and holiday greetings, and an array of the "neat stuff" that attracted visitors intrigued to learn how they had done it. A second and even deeper transformation of relationships was effected as Vinciguerra's special education students taught others—including teachers as well as students from all over the school—how to use the computer.

It is still a mystery to me how one computer with disk drive and printer, one special ed high school class, and one computer illiterate teacher got together to form one of the most successful communications programs I have ever worked with, but that's what happened.

With the teacher learning along with the students, it put them on an equal footing. This resulted in some very mature behavior on their part and some healthy learning for me as well as for them. The kids were very quick to pick up computer skills and were completely willing to give their time to "teach the teacher" or anyone else who came into the room.

In our school there was a marked change in the student-teacher relationship early on, as students realized that they knew at least as much as the teachers; students were far less timid about approaching new programs than the adults in the building. They were totally unafraid of the machine from the first and seemed to have a natural affinity for the thing. When they started calling it affectionately "Tom" (for totally obedient machine), I knew they were on a far higher plateau of computer intelligence than this teacher.

The fact that I was a teacher on the pre-K level of computer literacy gave rise to an unfounded fear that my computer might become little more than a mini game arcade in the back of my room. To prevent this uneducational occurrence, I immediately formed my first computer rule: **no games.** Lo and behold, the first piece of software I received was a typing program in which the letters of the alphabet became aliens to be shot down with ever-increasing speed and accuracy (so much for rule number one). The students' interest in this game did not help to allay my fear but tended to intensify it. Yet, they **were** learning to find the letters on the keyboard with that game, and some of them were even getting pretty fast. Hmmm. . . I wondered. . .? Well, maybe once in a while a game wouldn't hurt.

I already had volunteered to work with a fellow teacher on the school newspaper, which up until then had been a dittoed affair of five or six pages read by most students only if it meant freedom from more undesirable tasks in the classroom. My students were willing to work on the newspaper when pressured, but we had no real direction and little hope of receiving contributions in the way of articles from other classrooms.

We acquired a word processing program with the hope of typing our class work and whatever articles we received for the newspaper. The **PFS Write** program seemed to be an excellent choice for ease of use with special students. The amount of reading in the program's instructions is limited to one or two words and editing text is not complex. Another plus was the fact that the program allowed our novice typists to type lines of any length which were neatly converted with the press of a button into short little columns just right for the newspaper.

The magic came with **Print Shop.** With the aid of the program, my students found they could create six foot banners, cards, posters, individualized stationery and signs, all of which looked as if they had just arrived from the professional printer. The printing blitz had begun! First they designed a banner and logo for the newspaper. Then came the fancy posters asking other classes to submit articles. As articles came in, my students started printing related graphics to go with the text. They designed and printed fancy borders for each page of text: hearts for Valentine's Day, leprechauns for March, fall leaves, spring flowers.

Suddenly somebody felt the newspaper needed its own stationery for correspondence between classes. The Explorer Troop in our building needed stationery, the principal needed stationery and it all came out of our room nicely printed and designed.

With all that writing paper, we decided we should write to somebody: we wrote to our local newspaper, to our friends and to friends of friends. We wrote many letters requesting guest speakers and requesting free materials from anywhere and everywhere. We wrote business letters, friendly letters, thank you notes, birthday greetings, and cards of invitation.

By the time the first issue of our newspaper hit the stands, there were daily visitors to our classroom to find out how we were printing all that "neat stuff" on our computer. My class gladly taught all visitors whatever they wished to know ...ng as the visitor promised to submit a news article for the next issue (smart little devils, don't you think?).

Our issues were blooming into 15 pages or so and the ideas for permanent columns kept rolling in: How about a column about the computer? What about honor roll? Why don't we have a special section about kids who are doing really well in school? And thus were born the continuing columns of "Apple Seeds," "Honor Roll," and "Student of the Month."

At first, some of our contributing authors used pen names or initials when submitting articles to the paper. When the publication proved to be a success and the articles were well received, those very same writers started asking for by-line status.

By now the principal had an ongoing column and kids from all over the building were turning in jokes and riddles. Somebody asked if the computer could make crossword puzzles. Somebody else found out. We borrowed a program and a self-appointed group of good spellers and clever thinkers became our crossword editors. This naturally led to a column called "Oops!" in which we dutifully printed our mistakes from the last issue. Great ideas kept rolling in. We charged a nickel to run personal Valentine ads in February so we would have funds for a contest prize for the next issue.

Although students from all classes were submitting articles and participating in the production of our paper, the full-time permanent staff (my class) did the largest portion of the work. They labored without complaint and congratulated themselves daily on their own success. Self-esteem had grown to almost head-swelling proportions with one student in particular who displayed a sign on his desk which said:

Computer Problems/Any Problems
Need Help with an Article
Ask Bob

"Student of the Month," a column which started out with reviews like, "My God, I'd die if they wrote about me," later became a most sought after title. The honor of being chosen as "Student of the Month" included an interview with the newspaper staff, a letter of commendation in the permanent file, a certificate of merit in a leatherette case presented by the principal, and the name of the student displayed in the main office.

The work on the paper was becoming extensive and a need arose to set up editors from other classes to proofread and edit. Typists were recruited from everywhere. Some of the students who had taken printing classes in our shop program started to do layout for the paper. More and more people were taking part. School problems were aired in print and not a single guest got into the building without someone writing something about it.

Kids who couldn't spell found kids who could to work with them on the computer. Kids who could edit were working with kids who couldn't. They made bargains to trade skills ("Will you print me a birthday card for my mom if I proofread your article?"). We had kids who were counters, distributors, editors, reporters. Everybody had some part in the process. The only thing missing was a newspaper wire service, and you know what? With a little program called **Newsroom** and a modem, that's on our agenda for next year.

When we were not close to deadline on the paper, we wrote "Help Wanted" ads and "Situation Wanted" ads to put in our local town newspapers to help us get summer jobs. Each student designed suitable stationery and printed a resume to take to job interviews. Individual birthday cards, banners bearing names of rock groups or sweethearts had to be done on their own time. For this we set up a table for the "lunch bunch"—kids who wolfed down sandwiches in record time in order to have free time at the keyboard.

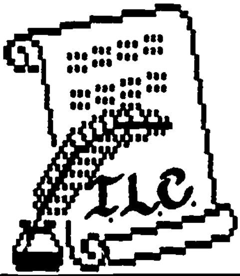
Several students who were particularly successful at teaching other students to use the computer were awarded a badge bearing the title "Computer Expert." Months later, these badges are still being displayed on shirts, belts and notebooks. One student recently dropped by my class for a replacement. He was concerned that his badge was worn and too difficult to read.

What about math and social studies and all the other subjects in my room? Well they're doing just fine, thank you. You see, you can't work on the computer unless all your other work is up to date. . . I used to write weekly behavior reports. Now I just turn on "Tom."

No description of this program would be complete without a few sample pages taken from actual issues of "What's News." Maybe they will encourage you to start your program today.

Sharon Vinciguerra has been a special education teacher for B.O.C.E.S. II Suffolk County for the last fifteen years. She has worked with students of all ages and is currently working with emotionally disturbed high school students. Her first experiences using computers for instruction and motivation have led her to several others. For further information on her program, contact:

Sharon Vinciguerra
1251 Walnut Avenue
Bohemia, NY 11716



WHAT'S NEWS?

SPECIAL EDITION

WHAT'S NEWS???.... THAT'S WHAT WE WANT TO KNOW! I.L.C. HAS AN AWARD WINNING (OR SHOULD BE) NEWSPAPER THAT'S GETTING READY TO ROLL. ALL WE NEED IS FOR YOU TO GIVE US YOUR IDEAS, ARTICLES, AND HELP AND WE WILL START THE PRESSES PRINTING.

WHO? EVERYBODY
WHAT? PUZZLES, ARTICLES, IDEAS, SUGGESTIONS, VIDEO/MOVIE REVIEWS AND SUGGESTIONS
WHEN? NOW AND THROUGHOUT THE YEAR
WHERE? W-9 & W-3

FOR ALL OUR NEW STUDENTS AND THOSE OF YOU WHO JUST FORGOT... HERE IS A RUNDOWN OF SOME OF THE FEATURES FROM LAST YEARS PAPER.

PUZZLE PAGE

A PLACE FOR CONTESTS, CROSSWORDS, MAZES, BRAIN TEASERS, OR ANYTHING THAT MIGHT BE FUN OR A CHALLENGE. SEND US YOUR IDEAS!!!!!

SHY?

IF YOU HAVE AN IDEA BUT ARE TOO SHY TO PRINT YOUR NAME, USE A PEN NAME. LOTS OF FAMOUS PEOPLE DO. LOOK AT THE UNKNOWN COMIC. HE SPENT YEARS WITH A BROWN PAPER BAG OVER HIS HEAD.

WHAT'S NEWS

SPORTS

TELL IT LIKE IT IS! BASKETBALL, FOOTBALL, TIDDLY WINKS IF YOU LIKE... AT SCHOOL OR IN THE PROS... WRITE IT DOWN

STUDENT OF THE MONTH

YOU OR YOUR TEACHERS MAY RECOMMEND ANYONE FOR STUDENT OF THE MONTH. THE STUDENT CHOSEN WILL RECEIVE A CERTIFICATE OF MERIT AND A LETTER IN HIS OR HER PERMANENT FILE.

WHAT'S NEWS STAFF

WE NEED REPORTERS, TYPISTS, EDITORS, PROOF READERS, AND MOST OF ALL; PEOPLE TO SEND IN THEIR IDEAS. WHAT DO YOU WANT TO SEE IN YOUR PAPER? LET US KNOW. WHAT DO YOU THINK OF:

1) A HOROSCOPE COLUMN?

2) A "MISS MANNERS COLUMN"?

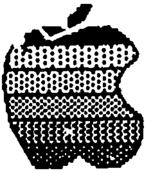
3) AN ADIVCE COLUMN (DEAR ANN LANDERS)?

4) YOUR STORIES, ARTILLES, POEMS!!?

V.R.C.

(VIDEO REVIEW CORNER)

DO YOU HAVE A FAVORITE MUSIC VIDEO OR MOVIE ? REVIEW IT FOR THE REST OF US.



APPLE SEEDS

A COLUMN THAT TALKS ABOUT COMPUTERS.

-TELL US WHAT YOU ARE DOING OR HAVE DONE ON A COMPUTER.

-TELL US WHAT YOU WOULD LIKE TU DO WITH COMPUTERS IN SCHOOL.

-REVIEW COMPUTER PROGRAMS.

S.G.O. NEWS

WRITTEN BY S.G.O. MEMBERS TO TELL EVERYONE WHAT IS GOING ON IN THE ORGANIZATION.

REFERENCES

Print Shop, Broderbund Software, 17 Paul Drive, San Rafael, CA 94903. Designs banners, letterheads, borders—prints 60 illustrations with a second disk with 120 more illustrations.

PFS: Write, Software Publishing Corporation, 1901 Landings Drive, Mt. View, CA 94043. Word processes: inserts, moves, erases, changes text without the need to recopy. Turns out crisp copies and stores info on disks.

Crossword Magic, L & S Computerware, Sunnyvale, CA. Automatically sets your words and clues in crossword form.

The Newsroom, Springboard Software, Inc., 7807 Creekridge Circle, Minneapolis, MN 55435.

WORD PROCESSING—ANOTHER VIEW FROM SERENDIP

by Sherry Anderson and Jack Halapin

Editors' note: The discovery that their seventh and eighth graders could not only edit magazine articles but also produce clean final copy for a literary magazine with the use of a word processor came as a happy surprise to Sherry Anderson and Jack Halapin—especially since most of the students had never used word processors, few could type at all, the job of choosing among the articles submitted still lay ahead, and only four weeks remained before the publication deadline. This account of their adventures, including the insights and adaptations necessary to make the process work, does make it clear that the success was no accident. It required considerable inventiveness and a certain amount of nerve, considering that Anderson "knew next to nothing about computers or word processors" when she began.

The word "serendipity"—which refers to unexpected happy discoveries—comes to us from a fairy tale, "The Three Princes of Serendip." Serendip is the ancient name for Sri Lanka, the island country in the Indian Ocean. The princes in the tale are continually making happy discoveries quite by accident.

Many of us may not readily associate serendipitous adventures with education. Yet when we put young minds together with microcomputers, the result is often educational serendipity—perhaps because computers in schools are still quite new. This opens the way to grand adventure, and one recent such adventure involved twenty-eight seventh and eighth grade students at the East Ridge Middle School in Ridgefield, Connecticut and their school literary magazine.

The setting was the school library. Nine Apple computers reside in the library in a room dubbed the "Apple Orchard." Their use and supervision during the school day is handled by the school's library media specialists. Computer arts classes, interdisciplinary classes, individual students, and teachers flow in and out of the "Apple Orchard" before, during, and after school, signing up for blocks of time for individual computers.

Computers have been a part of the East Ridge program for over two years. Computer Literacy experience is offered to all students as a part of the regular program of instruction. In addition, the computers are used for problem solving and for a limited amount of computer assisted instruction. But could they be used to help produce the school literary magazine? That question pointed the way to a serendipitous adventure.

COMPUTERS IN THE CLASSROOM

Doorways is the literary magazine produced by students at East Ridge. Its purpose is to give students a chance to see their writing published. It is a fine magazine, but this computerized issue would have a special significance. Twenty-eight boys and girls between the ages of 12 and 14 composed an "editorial board" and volunteered to undertake the effort to produce the magazine. These students were originally identified and nominated by their English teachers as having good writing ability, interest, and responsibility. This "editorial board" had the job of reading, evaluating, selecting, editing and proofreading student writing that had been collected since the beginning of the school year. This activity was, by design, an after school activity to be accomplished in a four-week period in April. It was a concentrated effort but one that had been accomplished in years past. This time frame allowed the magazine to be typeset, illustrated, printed and distributed before the end of school in June.

These elements came together this year with an idea and willingness to try something that had never been done here before. Could we produce not only the magazine articles but also the typed final copy, using a word processor? Heretofore the students read and selected manuscripts, but the job of producing clean, typed copy had been given to parent volunteers or teacher aides. Would a word processor, we wondered, enable the students to prepare the copy themselves and in the process to function more fully as editors—using a "real-world" tool?

The idea seemed reasonable. We needed to get clean, error-free copy for the printer. We wanted to provide the students with as full an editorial experience as possible, and using a word processor would certainly add an extra dimension. The students themselves liked the idea. We had the computers. And weren't computers supposed to save time? Perhaps the four-week work deadline could be met with somewhat less frenzy and worry than usual.

The problems were: (1) Most of the students who were going to do this had never used a word processor, and only the eighth graders had some typing experience; (2) The tasks of reading, evaluating, and selecting still had to be done; (3) The time for the four-week production had arrived.

First we needed a word processor. Actually, we needed a piece of software that would transform the Apple computers into word processors. (To make a computer a word processing device you need a program that tells the computer to allow you to enter text from the keyboard and allow you to review it and make any changes you need, for example, adding or deleting words or correcting spelling. It also lets you save the text and print the final document.)

There are many word processing programs on the market. But unfortunately many of them are difficult to learn to use without hours of practice. Selecting a word processing program that was easy for young children to learn and use would have been a difficult task. However, recently a software program called the **Bank Street Writer,** designed to be used in the home and the school, had been produced. The **Bank Street Writer** was supposed to be easy to learn and use. It had received some favorable reviews but, nevertheless, it still had to be tried out.

Using a software program with students is the best way of evaluating it; so the next step was to find some students and show them how to use it. We happened to have two boys who had used a word processing program before. When they heard that we wanted to produce the literary magazine on a word processor they volunteered to help, so we taught them first. This did not tell us what we needed to know, because almost immediately after a few instructions the students literally took off on their own and proceeded to enter text, edit it, and try out the functions of the program.

OK, it worked for these students. But what about the rest of our student editors; could they learn it?

Next, we selected two student editors who had never used a word processor before. We followed the same procedure. We put in the disk, turned on the Apple, loaded the program, and proceeded to show the students how to enter and edit text. We told them to copy this simple sentence, "Joe went to the store to buy a tuna fish sandwhich." Then we taught them how the program is used to change "tuna fish" to "ham" and to correct the spelling of "sandwhich." The rest of the training then involved adding more words to the saga of Joe's sandwich and editing, revising, and correcting.

Because the **Bank Street Writer** is clearly written and easy to use, the students had little or not trouble learning the program. Mistakes or confusion were easily remedied, either by the students themselves by trial and error, reading the manual, or by getting help from our first two students who by now (15 minutes after they first had seen the program) had become proficient at what needed to be done and could easily show others.

Soon we had going the equivalent of an "Each one teach one" literacy program. If you knew how to use the word processing program, or had learned how to do something with it, you showed the person next to you.

We quickly determined that there was little point in using our practice sentence when we had piles of manuscripts that already had been read and selected for possible inclusion in the magazine. So we turned all of the nine Apples into word processors by simply booting each computer with the **Bank Street Writer** one at a time. The process of practice then became one of entering the real text that was supposed to be word processed. Because of the superior quality of the **Bank Street Writer,** this learning proceeded largely without adult intervention.

In all, after a few days most of the twenty-eight students were easily and efficiently entering and editing text, and very happy and excited about doing so, even though typing copy is hard work!

Something else was taking place as well. For we were getting more than clean copy for the printer, more than correct grammar, spelling, and punctuation. We were getting sound content, too—helped, we believe, by the use of a word processing program. As students typed, they would evaluate what they saw on the screen for style, structure, word choice, paragraphing, consistency, and effectiveness. Their questions about a particular word, sentence, or detail were shouted out to the advisor, who could move quickly from editor to editor; or students asked each other for opinions and advice on revision, addition, or clarification. They had become not just individual manuscript readers, but a real editorial team—committed to publishing a magazine of the highest quality.

Using a word processor had the advantage of allowing students to be simultaneously involved with, and distanced from, the handwritten manuscripts. As they typed, they became personally involved with another author's writing. Yet, seeing the writing take shape on the monitor allowed them the distance to be critical. In addition, the ease with which the **Bank Street Writer** allows correction and movement of words or whole chunks of text inspired the editors to consider possible revision and improvements.

An editor would choose a particular piece of writing either because of personal interest or because the editor felt it was one he or she could handle. While typing it on the computer's keyboard, the editor would correct spelling and punctuation. Small revisions in word choice or grammar could be made here, but any questionable revision or changes in sentences or structure required a conference with the author. Since the author was not available (this being an after-school activity), the editor would save the work on disk and call the author on the phone or confer the next day in school. After discussion with the author, the editor would (1) make the revision, (2) request that the advisor look at the manuscript and hold a conference with the author, (3) return the work to the "maybe" file for further evaluation, or (4) give the manuscript to another student editor for a second opinion.

For students who did not know how to type, word processing was slow work—but interestingly the best (most error-free) copy came from just these students. They took their time and did not make the mistakes a typewriter-trained typist makes, such as hitting unnecessary "returns" and using the shift key alone to capitalize. Unfortunately, to get capital letters on the Apple II+ with the **Bank Street Writer,** you need to use a three-key sequence.

The two students who previously had some word-processing experience became the managers of the disk and printing production; indeed, they soon knew a great deal about the **Bank Street Writer.** They continually instructed the other student editors on how to use the program and supervised the saving and printing of all documents. They also volunteered to do all the final edits and corrections.

What this group as a whole—twenty-eight capable but inexperienced students—did with word processing seems very simple and logical now, compared to when we began. We learned as we went, driven by a need to get the work done. At first, the question was: "Could it be done?" The initial answer: "Maybe, let's try it." After the project was completed, the answer became: "It's the only way." Moreover, the students were not the only ones for whom the project provided some unique and valuable learning experiences. Both the writers of this article participated in the project, and at the outset only one was really familiar with computers. But that has changed. To quote Sherry Anderson...

> "To tell the truth, I knew next to nothing about computers or word processors when we began. When I heard you had picked up the word processor for us, I envisioned a crate housing something the size of a washing machine. But I had heard of them, and rumors of printers, daisywheels and spelling correction programs...and that was what we needed. We had four weeks to go from 300 handwritten submissions to printer-ready typed and proof read copy. We did it in three weeks and four days. Two months ago I was among the 'computer phobic.' My hands would sweat whenever I would venture to 'play' with the computers at school. Tonight sitting at the dining room table in my own home, I have just completed these notes for you, Jack, using an Apple II with its own disk drive and an Epson printer duly attached. Tomorrow, I'm going to finish the rest of the **Applesoft** tutorial..."

Sherry Anderson teaches English at Ridgefield High School in Ridgefield, Connecticut. She previously taught at East Ridge Middle School where she was advisor for the magazine described in this article. Sherry is no longer a beginner at teaching with computers. Her successful experiences have led her to integrate other software applications in her classroom.

Jack Halapin is Computer Facilitator for the Ridgefield Public Schools; he works with teachers and students at all levels to promote understanding and use of computers in the school and the world. We are certain that Jack's work with Sherry Anderson is only one of his success stories. For further information on their word processing program, contact.

COMPUTERS IN THE CLASSROOM

Sherry Anderson and Jack Halapin
Ridgefield Public Schools
Box 629
Ridgefield, CT 06877

ELECTRONIC NEWSPAPER: ONLINE WITH THURSTON

by Roxanne Baxter Mendrinos

Editors' note: At the suggestion of the library media specialist, teachers and students at Roxanne Mendrinos' junior high school organized an "electronic newspaper" through which they exchanged articles, stories, poems, editorials, crossword puzzles, and other features with students from two other Massachusetts schools. With a real audience for their work, students were strongly motivated not only to write contributions but also to edit and correct each others' work. Within the general excitement, several individual stories stand out, including the case of an emotionally handicapped boy who found in the newspaper a way to express himself—not only creating crossword puzzles, but even writing poetry.

The Edmund W. Thurston (EWT) Junior High School in Westwood, Massachusetts has initiated an electronic newspaper on microcomputer disk. This allows our students to interact with students from two other neighboring schools who can complete stories and answer questions posed by Westwood students, as well as share their original compositions and school news.

The idea for the electronic newspaper came from the library media specialist. Previously, our school did not have a newspaper or literary journal. The library media specialist approached the English and special needs teachers to enlist their support for the project. The faculty was intrigued with the concept of communicating with other schools through the use of floppy disk and the idea presented several potential benefits. It would reinforce word processing which is taught in the majority of English classes and in the special needs resource room. The electronic newspaper would acquaint students with the ease of word processing in organizing, rearranging, and manipulating text. The facility of editing copy was embraced by the staff as fulfilling the objectives of proofreading and correcting one's work. Students would make new acquaintances and share ideas through interactive technology.

Students were approached by their English teachers and by the library media specialist to be on the staff. At the first meeting, students decided what they wanted to include in the newspaper. Creative writing in the form of stories, poems, an editorial, sports coverage of a national and local nature which would not be outdated, crossword puzzles, horoscopes, jokes, and school events were included. An interactive story to be completed by participating schools and a question of the month would open the lines of technological communication. Students at this level enjoy branching stories in the form of "choose your own adventure." Students suggested beginning a mystery and sending the unfinished manuscript on a disk for another school to complete. The library media specialist contacted the computer coordinator from TEC (the Technical Education Cooperative), a cooperative education agency working with a group of neighboring member towns, and informed her of EWT's desire to find a school system using the **Bank Street Writer** to interact with through word processing software. Dover-Sherborn Junior High School was identified. The library media specialist then called and sold the idea to Dover-Sherborn's computer instructor. Sudbury Junior High School also responded positively. Contact was made by the library media specialist during a graduate computer course.

It was decided that contributions for the newspaper would be solicited from the entire student population, including special needs students. The student staff would be responsible for word processing all the material and formatting the hard copy. Enthusiasm was generated among the students by having a "Name the Newspaper Contest."

Students inspired their peers to write or submit work to the newspaper. Communicating with other schools, interacting on a microcomputer disk, and using a word processing program stimulated interest among the seventh and eighth graders. Teachers asked students to submit compositions of exceptional quality. The newspaper was receiving attention and support from both students and faculty. Staff support is extremely important in any endeavor. Student peer acceptance and enthusiasm is imperative at the junior high level for a successful venture.

One eighth grade English teacher cooperated with the library media specialist as senior editor and corrected the first drafts of the work. The first drafts were then randomly distributed among the staff to be typed into the computer and saved on disk. Corrections were made by the students using the editing features of the word processing program, **Bank Street Writer.** Articles which became out of date before the publication date were easily updated with little effort.

The library media specialist proofread the computer disk and corrected the errors. However, not all errors were detected by proofreading the copy on the computer screen. The work was printed on the proportional print mode of the Panasonic printer in a forty column display. It appears as condensed print so that the hard copy is assembled into two columns on one page. Students were given the hard copy to edit and correct for errors which had slipped through. Students retrieved the respective articles and made corrections to the text in a matter of minutes. Editing and manipulating text was exciting and efficient using the word processor.

Articles were then reprinted and cut by the students. The order of the articles was determined by the students. Articles were pasted on 9″ x 11″ paper by the students. Student artists were solicited by the English advisor and the library media specialist to read and illustrate some of the work. One student offered the software program **Fontrix** to design the cover lettering. The program has several fonts including Roman and Old English lettering.

When all the work was cut and pasted by the students after school, the articles were then assembled into the order they would appear in the newspaper. Several students were chosen to enter files in order into the memory of the computer so that the electronic newspaper would only be two files on the **Bank Street Writer** data disk. Each file was loaded into the memory of the computer from the original disk in the order that it would appear in the hard copy. Files were merged in this way. Copies of the disk were made to safeguard the work from a faulty disk. A data disk for **Crossword Magic** also was sent to the schools. The cooperating schools could also print out the crossword puzzles or play them on the disk. Both schools did not want to receive the hard copy of the literary publication before letting their students manipulate and read the disk. They wanted their students to print out the final copy. The hard copy was sent after the students from the neighboring schools viewed the disk.

Colored xerox paper from the superintendent's office was used. Students decided what colors they wanted to use to differentiate each section. The first page included an explanation of the electronic newspaper, an editorial and the "Question of the Month." The subsequent pages included creative writing and began with contributions on the meaning of friendship. This topic was chosen first because students were reaching out, hoping to make new friends. Poems followed with entertaining jokes, sports, horoscopes, and crossword puzzles completing the publication.

The library media specialist delivered the data disk to Dover-Sherborn Junior High School and gave it to the computer coordinator who then worked with the English teacher and the students involved in their literary publication to complete the interactive stories. The data disk was also given to the Sudbury Junior High School computer teacher during the graduate course. Students at Sudbury learn word processing in their computer literacy program. They worked on the disk and completed the story begun by EWT's students and included original compositions of their own for our students to read. An example follows.

FROM: DOVER SHERBORN
MYSTERY AT THE OLD CAVERN INN

by Laura Christian

COMPUTERS IN THE CLASSROOM

You are a famous undercover detective on vacation staying at a motel called the "Old Cavern Inn." You were hoping to get away from crime and cases and have a relaxing time. Unfortunately, the owner of the Inn, Mrs. Harper, is kidnapped. There are only three people who might have committed the crime: Philip, the gardener, Pierre, the bellhop, and Catherine Caper, a famous country singer staying for the weekend.

You interview each of them, hoping to draw some information from each. Philip has been working for Mrs. Harper since her husband died ten years ago. Philip never really liked Mrs. Harper and she doesn't trust him since they played a poker game which she won. Philip could never pay back the money he lost. Pierre was new at the Inn and had taken a liking to her. He is very upset when you try to question him, and throws a tantrum when you ask if he kidnapped her. Catherine, the singer, acted very snobbishly when you tried to ask her some questions. She is very disgusted about her stay at the Inn and, particularly, does not like Mrs. Harper. In fact, she hated her.

Who did it? Who committed the crime? You tell us by finishing the story!! Good luck!!

Edmund W. Thurston's students were very proud of their first enterprise. Twenty-two pages of the hard copy were collated by the staff and volunteer students after school and in the morning during reading period. The first page included an explanation of purpose "to give all students the ability to express themselves and show them how beautifully they can do it, too!" Donald Graves states, "It's true we can publish without a computer, but the capacity for making multiple copies and the clean-looking print will make it more seductive for the student to keep going, to keep writing, and publishing." (Green, John, 1984)

The enthusiasm from the students was overwhelming. Students wanted to be in the Library Media Center to work and socialize on the newspaper every day. They formed a close-knit unit with more students asking to be part of the staff. It was the decision of the senior editors to limit the staff to those already involved for fear it would become unmanageable. All students were encouraged to contribute, and all have felt part of the publication.

For example, one special needs student is part of the staff. Jeff is an average student with emotional problems, but he also has some literary ability. In order to encourage his positive gifts, he was put on the staff of the newspaper. He created the crossword puzzles, working with an outstanding student. He also expressed himself beautifully in poetry. Jeff has edited his work on the **Bank Street Writer** and is constantly in the library media center writing and typing in other students' work. According to his parents, it has been an important positive accomplishment in his life at the junior high level.

Denise is a very bright but timid student. When approached by her English teacher to participate on the staff of the newspaper, she was frightened and intimidated. Denise was insecure with the computer and did not enjoy BASIC programming. However, she enjoys writing and expressing herself. After the library media specialist taught Denise's class the **Bank Street Writer,** Denise was encouraged to use the word processor. She learned quickly and has enjoyed being part of the staff. She no longer feels self-conscious and has written and edited work on the computer.

Students have been extremely positive and proud of their endeavor. They have a feeling of importance without being egocentric. Both seventh and eighth graders, boys and girls, work closely without bitterness or rivalry, assisting each other on the word processor and in close cooperation inspiring their peers to become involved. Special needs students have found acceptance and a beautiful medium in which to express themselves. Students have never ridiculed anyone's work in the editing process. They have been responsive, caring, and enthusiastic about their participation, and their attitudes have influenced the school environment. It has been a rewarding experience for both senior editors as well.

"Composing with a word processing system simplifies the process of revision... children enjoy writing more and view it as a dynamic, speech-like activity that they can use for expressing themselves.... Word processing can make writing more fun and aesthetically satisfying simply because students can erase all traces of failure as they delete wrong words from the screen.... They can incorporate easily whatever final revisions a teacher or peers might suggest, without the tedious task of recopying." (Daiute, Colette, 1983) This, too, has been our experience.

Students from the Sudbury Middle School wrote to EWT's staff, "On behalf of the students at Curtis Middle School, we thank you for putting forth much effort into your computer writings. We were fortunate to receive your literature and were very impressed with the articles. Your motivation stimulated us to write our own stories, and we've had much enjoyment..." Westwood's students hope to meet their neighboring peers in person before the end of the year.

Roxanne Baxter Mendrinos is Director of Library Media and Computer Applications at the Edmund W. Thurston Junior High School in Westwood, Massachusetts. Experiences with computers and online databases during graduate school have encouraged her to develop many classroom activities. The philosophy of *her school is to integrate the computer as a tool within the curriculum. Her upcoming projects include several applications of* ***Appleworks*** *and satellite telecommunications. For further information on her electronic newspaper, contact:*

Roxanne Baxter Mendrinos
Edmund W. Thurston Junior High School
850 High Street
Westwood, MA 02090

REFERENCES

Daiute, Colette. 1983. Computers and the Teaching of Writing. **On Computers and Writing.** Risdon Publishing Co.

Green, John O. March 1984. An Interview with Donald Graves. **Classroom Computer Learning.**

COMPOSING WITH THE WORD PROCESSOR

by Carolyn Kohli

Editors' note: With students already trained in word processing by her school's business department and a computer for every pair of students in her class, Carolyn Kohli could concentrate on teaching writing rather than word processing. She created a writing workshop. The word processor made revisions far less laborious, and Kohli invented a variety of techniques to take advantage of the capability, techniques involving students' critiquing each others' work, as well as techniques for using the machines directly. Some compromises with her original vision of students writing as well as revising on the machines proved necessary for practical reasons, but she devised ways of coping with the limitations—in many cases, making a virture of necessity.

Claudia Ann studied the line she was composing and changed **walk all over** to **step on.** Then **step on** to **jump on** to **smash** to, finally, **tap dance inside,** and that was it, the image she wanted to convey. Now the final version of her poem read

When I found out my best friend had been
Talking about me

I wanted to do a tap dance inside her brain until
Every little rotten piece came running out
And down the side of her face

I could have dived into her skin
And wrinkled up every bone
In her body

I felt so hurt and disappointed.

(Claudia Ann Williams)

Her classmates' work was equally vivid. One student described a sister this way: "like a visitor, you come home when you have no other place to go." Of arguments between friends, another wrote, "cruel words spoken back and forth sharp as a knife/ like the fire dancing in front of the moth." And a third wrote that falling in love made her "joyous plaid." The poems were good because the poets were spending time rewriting, trying different words and phrases and deleting and inserting lines until the verses looked and sounded the way they wanted them to. This was no ordinary creative writing class, though, comprised of bright kids who love to write. These were fair-to-average students who were struggling with imagery and figures of speech, line and form, but doing it all on word processors. And that made the class new and exciting.

It seems that everyone's jumping on this computer band-wagon, though some may see it as a juggernaut. Programs to teach sentence structure or to prepare for the verbal section of the SAT are now available, and it appears that every school in the country will have at least one computer lab before the decade ends. So teachers of English, no matter how skeptical, should seriously look into the possibilities of teaching writing with the word processor. I was not particularly impressed at first and felt that word processing, like typing, was a handy skill, but what place did it have in the English curriculum? Emerson's "things are in the saddle and ride mankind" came to mind: here it was, just another thing with which we'd be saddled. Then I had the opportunity to play around on a computer. I wasn't an immediate convert, but I did agree to help develop a class using the word processor to teach writing.

Initially, I wasn't sure that the course as envisioned by the school administration would work. First of all, I thought word processing should remain under the aegis of the Business Education Department, but the principal wanted to try a class in composing with the word processor. We began by agreeing that the focus of the course would be the writing, not the machine. Then we decided that Composing with the Word Processor would be an elective with a preprequisite of a course entitled Introduction to Word Processing taught by the business department. As a result, the first students arrived already motivated because they'd elected to take the course. And since they knew the rudiments of word processing and typed at least 40 words per minute, it really was a writing rather than word processing class. We determined the class would be like a laboratory, a true writing workshop where students would do all composing in class, but we would include literature, too, as writing and reading go hand-in-hand, so essays and poetry were read and discussed as models for writing, while a novel and biography were read outside of class for a literature essay. Writing was the first priority; reading fed the writing.

The early curriculum included poetry, a first-person narrative inspired by James Thurber's "University Days," a critique of a school play, a literature essay based on outside reading, and a research paper. Thanks to a colleague who also believed the machine could be used to teach, we added concrete poems or "emblems" to the poetry section. This colleague, Win Radigan, put e.e. cummings' "r-p-o-h-e-s-s-a-g-r" on the program disks and had her students call it up on their screens. Then they wrote the poem's ideas in prose prior to composing a similar emblem. One student created this missile:

t
hek
illin
gmachin
esherethe
ycomecan't
bestoppeda
fterthey're
launchedw
illtherebe
atomorrow
toremembe
rouryester
dayswillthe
rebeanendt
othischildish
gamewepl
ayhowwillit
endbyagree
mentorby
ahugebla
ckcloudtel
lingusit's
al
lover
formankind
) & !
$ / ?
* + (
(Luis Cortes)

Another writer, whose subject was birds, spread words across the screen to imitate a bird's flight. It was fun to see what students could do with imagination and a screen on which letters could be arranged and rearranged and centered and capitalized and moved up or down without ever having to scratch through and start again.

To encourage rethinking and rewriting, I prepared a lesson on the disks that showed two opening paragraphs of a student essay based on the Thurber lesson. The students saw this on their screens:

-A-

The class I disliked most was an English class in junior high. Mr. Sodden was the teacher's name. His class was at 12:30 in room 320. He never taught a lesson. He only handed out crossword puzzles. Mr. Sodden had no control over the class. They threw paper airplanes in class.

-B-

Total chaos results in a classroom when a teacher has absolutely no power over the students. Each day, five days a week, chaos reigned in room 320. As paper airplanes would whirl through the air past Mr. Sodden's ear, he would continue to hand out his crossword puzzles.

(Arlene Donnelly, "Whirly Words")

The students answered questions about Arlene's revisions, then using the command for making columns, they typed in the opening paragraph of the narrative they'd just written in the first column and rewrote it in the second. In most cases, the revision was superior to the original. This was getting to be fun and productive, so we went on to prepare book report questionnaires and notetaking instructions for literature essays on disk. The students enjoy answering questions this way; it also provides them with notes prior to writing expository material such as reviews and analyses. Consequently, their writing is better.

None of this is very difficult to accomplish in any school, nor does it take an extraordinary personality to pull it off. It does take creative teachers of English who are willing to experiment. I found, for example, that my role and the way I spend class time has changed dramatically in Composing with the Word Processor. I teach a formal lesson—that is, bring in a piece of literature, stand in front of class leading a discussion, have students take notes on the computer, and give the assignment—about once every ten class meetings. The rest of the time, I act as a sort of roving editor or personal tutor. I help students get started or overcome momentary blocks; spell words or direct them to the dictionary and thesaurus; ask writers to read awkward or poorly-constructed sentences aloud so they can hear what I see on the screen; and sit by each person while he or she composes.

It is a pleasure to be able to work on individual writing problems while the student is writing. Stephanie, a girl who wrote run-on sentences chronically, began to read aloud to me, and each time she stopped naturally, I'd have her insert a period or conjunction. Eventually, she was reading her work aloud under her breath and correcting the run-ons herself. In regular classes, we often grab a student for one minute before or after class to explain comments we've written on a paper. In this class, time for that was built in. That's one reason a girl like Stephanie could become a better writer. That she could make corrections quickly and neatly was a factor, too.

As many people have noted, the word processor makes writing and correcting so easy that it's a great motivator; in the case of high school students, this feature should not be derided too quickly. Of course the machine will not teach the skills one needs to write well, the skills of clear thinking and comprehensive reading that lead to effective writing. But because it makes composing and revision less tedious than the pen-and-paper, or even typewriter, method, student writing tends to be better. If I hadn't known it before, that first class taught me that most kids scrawl off essay assignments and turn them in unedited. The word-processing students did better work because we could talk in class, they could edit while I looked on, and they could correct from a pencil-edited first print-out very quickly and easily. And because the writing is neatly aligned and printed, on the screen and in "hard copy" (on paper), it's very easy to read. Only a person who's never faced a weekend of marking three or four sets of essays would say that's a petty concern. This is definitely a benefit derived from technology.

The English chairman of another school who was visiting one of my classes recently arrived just as I discovered that the power to ten computers had been turned off; so he was treated to my scrounging around in the bottom of a cabinet digging out extension cords that we keep for such an emergency. I've never met anyone so pleased to see things fouled up, but he smiled through the entire visit because he'd had a chance to see this "exemplary program" in a troubled moment. Well, there were a number of problems with Composing with the Word Processor, of which accidental power loss was only one.

In the first place, we had 17 TRS-80 Model 4 computers, not a bad machine, nor is its word processing program, **Superscripsit.** But we had 34 students in each section, so I worried about how to rotate time spent on the computers and what non-writers would do. At first, I decided the non-writing student could look over the writer's shoulder and offer suggestions, or at least point out spelling errors. Personally, I'd dislike such a shadow, but it seemed justifiable. The reality was a surprise: students paired up at the machines on their own, friends with friends, and rotated time fairly and naturally. Non-writers often began a rough draft in longhand, not my original idea of composing from beginning to end on the word processor, but they were anxious to save time. Later on, I placed them in reading groups of four each with a worksheet and had them read their compositions aloud and criticize them. This activity became a part of each assignment, so in addition to turning in pencil-edited and final drafts of their writing, the students also had to attach the comment sheet from their group.

No matter how we plan for the use of class time, however, reality and personality twist a knot into every scheme. There were always a few students with nothing but time on their hands, so if they didn't have the book they were reading for the literature essay, I assigned them extra essays to read in an anthology. Then, there were kids who were better at working the computer than others, so when they had completed an assignment, they helped those who were having trouble recalling commands or getting the program to work. Unfortunately, there was one time even they couldn't help, and I found that I was spending most of my time helping students with commands rather than commenting on writing, but that was a result of a poorly-planned unit which I am eliminating next time around.

And then there are the machines themselves, which cause more trouble than people or lesson-planning. On hot or muggy days, the machines overheat when the air conditioner in the lab doesn't work. We had a week or more of that, and kids simply had to write longhand or read, a real waste of lab time. Sometimes a program disk begins to "kill itself off," as this mysterious sudden malfunction was explained to me, leaving a student in mid-composition with crazy hex symbols on the screen or work completely erased.

So we had our share of frustrations and failures. But none of these problems warrant the kind of blanket condemnation of word processing in the English curriculum that I've heard and seen in print lately. Some Cassandras are crying that these new computer labs are ill-conceived, expensive, trendy parallels to the language laboratories of the 1960s, doomed to the same wasteful end, and, as I understand it, the basic problems are the same: vandalized equipment, uninspired or disinterested teachers, and budget concerns.

The computers we used in Composing with the Word Processor were delivered to the school last June, and nearly every one has been back to the company for repair at least once since then. That meant we were without machines for two or three weeks at a time. Three already have been vandalized: students scratched epithets in two and stuffed sunflower seeds in a disk drive of another. Disks have been damaged by overuse or mishandling, and they've been lost or stolen. Insofar as staff is concerned, only two of us expressed an interest when the course was first proposed, and three of us are able to use the word processor. So it would seem the nay-sayers have plenty of fuel for their negative comparison, but I don't see mountains of broken personal computers stacked up in supply closets. The problems may be perennial, but they're not insurmountable.

Consider typing classes: typewriters break down regularly, but typing is still taught in public schools because it's regarded as a skill worth learning, for college-bound as well as secretarial students. Why should word processing be different? Also, as my chairman, a rabid anti-machinist says, computers aren't coming, they're here. Furthermore, being computer literate is important to us; I don't think being multilingual was. Finally, once students learn to operate the machine, all we have to do is supply the inspiration through the writers we love best, and the kids will com-
: and revise because it's not such a burden. That, not self-correcting work

book exercises or vocabulary quizzes, is the real power of using the computer in the English curriculum.

Carolyn Kohli, who teaches at the Edward R. Murrow High School in Brooklyn, New York, helped to create and teach the course described in her article. Carolyn confesses that prior to developing the course she had successfully avoided computers in the classroom. Her school administration is supportive and encourages experimentation. Under these conditions she has continued to develop ways to use computers as tools for learning. For further information on composing with the word processor, contact:

Carolyn Kohli
41 East 3rd Street
Brooklyn, NY 11218

THE CRAFT OF PROCESSED WORDS

by Michael O. Stegman

Editors' note: Apart from the great pleasure of publishing an article about teaching writing that is itself superbly written, we selected Stegman's article for the inventiveness of his exercises designed to teach the use of four essential word processing functions: creating, adding, deleting, and moving text. Perhaps selected exercise titles will intrigue prospective readers: "Mental Jogging," "Ways to Avoid Lending Anything to Anyone," "Inner Thoughts of the Great," and "My Lying Self." Through these exercises, Stegman cultivates the skills necessary to a full realiztion of the computer's potential for writing instruction. Such explicit preparation through finger exercises emphasizes the indispensability of strategic interventions by the teacher if computer-based tools are genuinely to empower students and promote learning.

The term word processing does not adequately describe the role the computer can play in the writing process. For that term's implied stack of homogenized, sliced, and wrapped words defeats what we have learned about the ways in which students learn to write. Consequently, when I considered using this tool in my classroom, I began with the conviction that I would take the false promises of a generation of essays without mechanical errors and relegate them to the same position I hold for proofreading: the last. Instead, I wanted to see if I could devise writing experiences in which the rather limited functions of a word processing program would be an ally for sound writing instruction: this meant that I had to get beyond the computer's penchant for mindlessly repetitive tasks to mindfully repetitive ones.

Once I had come to this rather obvious insight, I realized that only four functions should concern me most: creating, adding, deleting, and moving text. I wanted the work that students accomplished at a computer terminal to encourage active involvement with language through playing with language in creative ways; and so, I set as a reasonable demand for myself that the assignments had to engage students immediately and intimately. Finally, the exercises I devised for the four functions of a word processor have a further agenda: to establish the foundation for another writing skill—revision. However, before students can take advantage of this feature, they must still find a way to get their words into the computer or, more circuitously, onto paper and then into the computer.

A computer's word processing program alone does little to initiate and shape the thoughts during the early stages of writing. Although the movement of the computer screen's cursor and other sensory cues may encourage students to write more, high school aged students may not be ready yet for freewriting at a computer terminal. The paucity of computers and the tendency for those present to be in a lab where thoughtful composing may be difficult at best both ensure this tool will not totally replace pen and paper. In most of the situations I know of, students bring handwritten essays to the computer lab, enter them, and then continue on to revision. These limitations cramp the computer's use as a natural extension of a student's writing experience.

Necessarily, students begin with short manageable tasks and move to larger ones that reinforce an environment for inventive revising. The assignments presented here do not constitute all of the writing done with and without the computer in my classes, only a selection. In the beginning, they also have a large element of play in them to sustain a student's enthusiasm and to encourage exploration of a number of different writing voices or points of view. Becoming comfortable with various voices also facilitates students' writing later when they assay ideas from more than one perspective to create their own meaning.

I. Creating Text

Because students in Shoreham-Wading River High School work with a Digital Equipment Corporation PDP-11/44 minicomputer and the LEX-11 word processing program, the first of these introductory assignments familiarizes students with the computer environment and with the basic structure of a word processing program. It takes about fifteen minutes to pass on the necessary information needed to log on to the computer and create a document to work in. Much of this is already known by students, and what isn't obvious is included in on-screen directions. The first assignments encourage the typing of ideas without worrying about proofreading. These writings are immediately shared with writing groups for positive reinforcement and to encourage more writing through imitation of other students' models.

Mental Jogging. The first exercise freely adapts and even uses (with permission) Reid Daitzman's book of the same title. Students choose from a list of topics that typically asks for ways to do something, reasons, or consequences. For example, seven or more things to do while waiting for your parachute to open, or seven or more ways to peel an egg. After listing the first three things, students usually run out of cliched responses and must now enter the arena of inventive thought—just where I want them to be.

Ways to Avoid Lending Anything to Anyone. After a reading of Adrian Mitchell's poem "Ten Ways to Avoid Lending Your Wheelbarrow to Anyone," students are asked to imitate the formula for this poem as they devise their own ways to avoid lending anything. The poem repeats a pattern of adjective, question, and response. The elements depend upon each other in ways that encourge playing with voice and diction. Here is a selection from the poem.

MACABRE
May I borrow your wheelbarrow?
It is full of blood.

A student response could run like this:

PARANOID
May I borrow your wheelbarrow?
Why, is there one following me?

PARANOID SCHIZOPHRENIC
May I borrow your wheelbarrow?
Why, is there one following us?

Because LEX-11, our word processing software, is programmable I also include the formula for this exercise as a series of prompts. In fact, I have set up all of these assignments, including helpful suggestions, on the computer. Granted, this benefit depends upon the particular computer configuration and software we use, but there are ways around this that a microcomputer will still allow for. After all, the assignments are based on the four major functions that all word processing programs can perform.

These are not the only assignments that I have devised for this element of using the computer in my classroom, but they represent the kind of writing that my students do to create text through listing. Later in the year when students gather material for a longer piece of writing, they will have already developed this skill for their writing repertory, as well as an awareness of how voice and diction can convey different meanings. To reinforce listing as a creative activity, I also salt the lessons with these and other poems: Paul Viola, "Index" in **Splurge**; Wallace Stevens, "Thirteen Ways of Looking at a Blackbird," F. Keith Wahle, "Apologies" in **Old Friends, New Friends**; Ron Overton, "Driving: Yes or No?" also in **Old Friends, New Friends**; Elizabeth Barrett Browning, "How do I love thee, . . ." and William Shakespeare, "Sonnet 130"—"My mistress' eyes are nothing like the sun."

II. Inserting Text

The second major function of any word processing program inserts text, and assignments to teach this function begin by supplying the text for a student to work with. Subsequent assignments require students first to create and then add to their own text, and the concluding assignment transfers this skill to commenting on a poem or other short work of literature in the manner of an on-screen reading journal, a journal writing technique our students use to comment on [illegible]r they read. In all cases, the function used, inserting text, also stresses voice and diction as a way of indicating the reactions and biases of either an assumed narrative voice or that of the student, a clear prelude to analysis. While this last goal may sound grandiloquent, the assignments are more concrete and decidedly popular.

Inner Thoughts of the Great. Provided with a list of famous and not-so-famous quotations, students must add their own speculations about the speaker's inner thoughts. In some cases the thoughts run counter to the professed mood of the piece. For example, one student used Lincoln's Gettysburg Address to reveal that Lincoln was in fact thinking more about theater tickets than he was about the memorial service he was speaking at. On a more serious note, one student imagined that in the pause between "and yet" and "it moves" Galileo reviewed the importance and personal consequences of his own denial.

My Lying Self. This assignment builds on an ability to manipulate voice and diction begun in "Inner Thoughts of the Great." This time around students are given a selection of dramatic situations with which they are either actually or imaginatively familiar, situations requiring that they explain something to someone else or excuse some action of their own. After making a choice, they write an explanation for this audience. When they have completed this, they then insert into that explanation what "really" happened. For example, students must explain how their family car got that dent near the left rear wheel.

Reader Response Writing. Students in the Shoreham-Wading River School District are accustomed to maintaining reading journals. In this assignment, instead of writing their responses in a separate journal, they add them to a literary text that is displayed on their terminal's screen. I provide them with a number of short texts to select from and they return to the text repeatedly as their writing groups encourage more detailed responses.

The following are examples of kinds of texts students are encouraged to work with: the encounter with Nature from the "Ktaadn" section of Thoreau's **The Maine Woods,** the pine forest section in the "Chesunkook" section of the same book, the discussion of the "Society for the Diffusion of Useful Knowledge" in Thoreau's essay "Walking," and from Pascal's **Pensees,** the sections on man as a thinking reed and the two infinities.

What appeals to me most with this assignment is that the responses remain imbedded in the text at all times, allowing for a constant dialogue. As the students write, they can accumulate specific evidence for their ideas by using quotation marks to annex a section of the author's text as their own. A side effect of grabbing text as one's own in this way facilitates the task of specific references and to include them in the natural flow of a student's ideas. This process revives a technique introduced in "My Lying Self" which intimated that students could blend information and commentary. Finally, although other methods of essay organization work well, responding directly to an author's ideas as they develop in the essay or story also reinforces a student's own growing awareness of a passage's meaning.

III. Deleting Text

The third major function a word processor performs, deleting text, proves harder to create valid assignments. The context for these assignments differs slightly from the previous functions, since it is so obviously related only to revising text and not to creating text. This series of exercises explores the benefits of manipulating an author's text and then one's own to decide on or discover the major issue of a piece of writing. The analytical task asks a student to recognize the major points of a work's thesis as it develops and becomes futher refined.

Abstract Concrete. This assignment shares some of the techniques of "Reader Response Writing," but adds one step. Instead of inserting responses into a piece of writing as before, students first look to reduce text from several screens to one screen. Often a group of students work together to arrive at a consensus of the text's major points. Once they have achieved this, through whatever criteria they devise, they expand the text by inserting into it their comments on each of the surviving points. Further, within this agreement an individual can still express individual preferences and nuances that others may not have seen. Because each point must be first agreed upon and then commented on, the writing here stands a good chance of holding more steadily to the ideas of the author.

IV. Moving Text

Admittedly, this fourth and last function of the word processor has almost successfully resisted my efforts to incorporate it into a writing sequence. After all, the problem it poses is this: how to design a writing assignment that naturally and inevitably leads a student to write meaningfully and then revise in such a way that the basic organizational pattern is altered. A student writing group can reinforce this process as they discuss another student's essay among themselves, but the computer alone cannot offer the kind of responsive interaction that students responding to each other's work can do. In fact, my work with the computer has shown me that its use is best yoked to facilitating the work of a writing group. With these two elements working together, the computer can easily and deftly implement the suggestions of a writing group, and the functions of a word processor can suggest ways in which to consider altering a text.

Pedwin. Students in my classes know before they arrive in class in September that one of the best things they will be doing during the year will be to perform in the school's "Coffee House" on the day before Christmas vacation. While everyone assumes they will at least extend the tradition of the "Samuel Beckett Memorial Kazoophony and Bus Stop," they also begin to plan for the yearly appearance of the works of E. Elmwich Pedwin.

My class invented Pedwin nine years ago to hoax the rest of the school. Yearly we take over display cases in the school library to mount a "shameless display" of his works. Along with the memorabilia comes the first performance at the Coffee House of yet another work of questionable literary merit from the exploded ball point pen of one of America's greatest living dead drunk poets. (As you may be able to detect, a long-standing tradition of jokes is associated with this project.) Generally, the classes decide upon some general theme for the yearly skit: one year they did an opera complete with playbill (or Playbob) and truly awful singing that floored the audience; another year they did a variation on **Waiting for Godot** in which parodies of faculty members wandered past Didi and Gogo; and recently they not only parodied a fairy tale, but treated it as a foreign movie with dubbed dialogue. No one wanted to memorize lines, so we recorded everything before the performance and students mouthed their lines hilariously.

The entire project is done on the computer with each class or groups within a class providing episodes for the play's story board. Somewhere near the end, we refine the play by shifting, adding, and deleting scenes. The process goes quickly, and we can easily try several arrangements because the computer does the cutting and pasting for us. During these revision sessions, I have access to a large projection screen that encourages group participation.

There is life beyond all of these assignments, since they only introduce concepts that the writing done in the rest of the year can rely on. I prefer to introduce writing with a computer as an intimate of the process of revision, the crucible for a student's ideas, instead of stressing it as the means for skimming off the dross of misspellings and punctuation. Further, these are not the only ways in which I employ the computer to encourage and assist student writing. Our system allows for bulletin boards, electronic mail, and other features that facilitate the sharing that a writing group also fosters. In one case, after students discover the pleasure of sending and receiving mail as well as the other announcements, they created an epistolary novel. For those who need work with editing, I use a projection screen and a computer terminal instead of an overhead transparency.

At the moment, the most intriguing project my students and I are involved in concerns how we can best use DECtalk, a sophisticated voice synthesizer. Already we've planned the obvious—a computer play using its ten programmed voices. In fact, the Advanced Placement English class has arranged for the voices to "speak" to the school's principal about his resistance to computers in the human-[illegible]. And it is not too difficult to see how the voices could help reinforce assignments such as "Inner Thoughts of the Great" and "My Lying Self." Further uses will extend it into the ESL classroom and for those students who would benefit from a disinterested reading aloud of their essay.

And so, the computer can go beyond the four basic functions of a word processor—creating, inserting, deleting and moving text. The assignments I have outlined, even those that ask students only to create text, have been aimed primarily at enhancing the possibilities a student might consider during the revising process. The tasks have deliberately stayed away from the trap of emphasizing the computer as an editing tool. Instead, I have preferred to rely on the larger unit of text as the vehicle for teaching the four word processing functions. I have done this because I most want to establish careful analysis and commentary as the primary goal within my students' writing and revising, and these assignments satisfy this goal for me.

Michael Stegman teaches English to twelfth graders in a rural high tech community on Long Island, a paradox he does not try to resolve. Mindful that the real purpose of processing words is to read them afterwards, he devotes himself to his goal of becoming the best possible audience for modern poetry and also serves as a trustee for the Shoreham-Wading River Public Library. To test the limits of computer use in an English classroom, he is now developing writing lessons that incorporate an interactive laser disk containing selections from The National Gallery of Art in Washington, DC. For further information on his program, contact:

Dr. Michael O. Stegman
Shoreham-Wading River High School
Route 25A
Shoreham, NY 11786

II LEARNING TO INQUIRE THROUGH DATABASE PROGRAMS

INTEGRATING DATABASE PROGRAMS ACROSS THE CURRICULUM

by Howard J. Moskowitz

Editors' note: Howard Moskowitz' article is a useful introduction to databases and their uses across the curriculum. Moskowitz defines the elements of a database, then explains how teachers and students in his district use database programs at three progressively more sophisticated levels: using an existing base of information, building their own files, and designing the data files from scratch. He provides examples at each level and includes recommendations of numerous commercially available data files and data management programs.

The decade of the 80's has been characterized as "The Information Age." Today our economy is based more on the creation and distribution of information than any other resource. According to John Naisbitt, "In an information society, value is increased by knowledge. The new and sophisticated information technology brings order to the chaos of information pollution and therefore gives value to data that would otherwise be useless. Technology will help us manage the information society only to the extent that its members are skilled in utilizing it." (Naisbitt, 1982)

This paper focuses on incorporating a flexible tool, a database program, into the daily instructional process. Change has always come slowly to schools; here is an opportunity to utilize something from outside of the schools, to help to improve instruction and ultimately increase student learning. Database programs and computers have become an integral part of the real world outside of the schools. The Toledo Public Schools saw a chance to integrate a real world activity into our classes to help us use the new technology with some of the instructional activities we have been doing for the past 20 years.

Below we describe some typical applications for a database, software we have used, and data files we have created within the Toledo Public Schools. But first, it is useful to think about the basic features of this type of software and what makes it a powerful and flexible tool. What is a database?

A database program is software which converts a computer into an electronic filing cabinet full of information. Without computers we have had to store information in notebooks, on filing cards, or on pieces of paper in manila folders. The computerized database allows organization and storage of information in an efficient manner, on a computer disk. Using a database provides the capabilities of quickly and easily searching and sorting through large amounts of information. In just a blink of an eye, we can retrieve specific information stored on the disk or make updates to the existing information in the database.

A **file** is used by a database program to organize and store information on a particular subject. A **file** is composed of individual **records.** The **records** are made up of specific categories of information called **fields.** A **field** is the smallest unit of information collected; it is usually a distinct category of information, such as "a student's first name," "last name," "street address," "phone number," and so on. We can easily design a file where each record contains several fields; one for each separate piece of information needed. For instance, in the administrative offices of our schools we use a simple file where each record contains five categories (fields) of information. With this type of file we are quickly able to locate basic information about particular students or whole classrooms. We have designed several database files both for administrative/managerial and instructional applications.

There are many instructional applications for databases. We have been able to take disjointed pieces of knowledge or facts and provide some cohesiveness to help students better understand the information. Students are able to actively interact with the information that is stored in the database. The information becomes more meaningful for the students; it is no longer just facts to be memorized and studied. There is an old proverb that states: "I listen and I hear; I read and I remember; I do and I understand." In essence, students become active participants in the learning process when they are manipulating database files. As students begin to explore databases in their classes they are involved with learning skills and activities that they may never have had an opportunity to use before. Some skills include: planning and development, collecting data, synthesis, analysis, comprehension, evaluation, classification, and categorization.

Database activities can be incorporated into just about any curriculum area. Compared with computer assisted instructional programs, databases are not content dependent. Most computer assisted instructional software is designed for a specific subject or concept and written for a specific grade level. The software limits what students are able to do. Teachers are dependent on locating the most appropriate software to match their specific curriculum and grade level.

Instructional activities in most curriculum areas can be developed or purchased for use with a database program. We have found that when students, as well as in-service teachers, use a database program they progress through three stages of involvement, or three levels of learning activities: using an existing file, building a file, and then designing a database file.

...e final step, the students use existing files that have already been created by someone else. Teachers create their own databases of information, relating to their specific curriculum areas, or commercial databases can be purchased. We use a series of excellent curriculum data files created by Scholastic, Inc. These include data files in United States history, United States government, life science and physical science. In order to use these files the database programs **PFS-File** and **PFS-Report** are needed.

Another very fine database series is being developed by Macro Systems, Inc. and Intentional Educations, Inc. Macro Systems is developing textbook-related data files to correlate with world history, and appropriate for learning disabled students. These files include a very powerful database program, which allows the students to easily manipu ͏ e the information. **Newsweek** magazine also has developed a series of data files, which use the **AppleWorks** database program.

Our students and teachers become actively involved in the learning process when they begin to manipulate the data contained within the database files. They begin to explore the data files and learn how the informtion is categorized, organized, and related. Students and teachers learn how to answer specific questions relating to the information stored within the files. Teachers learn how to ask specific questions relating to the data. Teachers have to rethink their previous organizational patterns for the content they are teaching to students. Both teachers and students manipulate, rearrange and finally print out the information which is contained in the data files. The manipulated information is used to answer specific problems and questions. Students are actively asking questions, defining procedures for solving the questions, and determining what information is needed to find an answer. The end result involves the students actually manipulating the data to develop an answer and to interpret the results.

Some activities used with the existing Inventions and Technology data file allow the students and teachers to determine what data is stored in the file. Questions are used which allow exploration to determine similarities or differences of the information stored in the file. How many inventions are related to agriculture, medicine? How many inventions were there prior to 1950, and what are they? How many women inventors are there?

For us, the second stage of activity involves the building of data files. Teachers must first develop a series of problems or questions relating to curriculum content. These problems must be global enough to lead the students to do some research in order to find the answers. The teachers provide the students with a file format and information requirements. Students then search for the data and enter it into the file. The ability and grade level of the student determines the specific content of the data collection form and the file form. A fifth grade ͏abase for American Presidents is not as involved as a high school file. The high school database will include more data and information. In particular, the fifth graders will not be learning about foreign policy; however, this topic is included in the high school file. After the file is created, the students can then manipulate the data and answer the problems.

Figure 1

Database Format:
Fibers and Their Characteristics:

Name of Fiber:

Type of Fiber (natural or man-made):

Source:

Origin:

Characteristics of the fiber:

Positive:

Negative:

Press Control -N for Next Page

Care Instructions:

Uses of this Fiber:

Additional Information:

References:

Press Control -P for Previous Page

Created by Karen Centers, Home Economics Teacher, Jones Junior High School, Toledo, Ohio.

FIGURE 2
SCHOLASTIC WORKSHEET
INTRODUCTORY ACTIVITIES

Selecting the SEARCH/UPDATE Function

The PFS File Function Menu is on your screen It lists six functions. The only function we'll use right now is the SEARCH/UPDATE function. This is the part of the *File* program you use to find information stored in a file on a data disk.

1. Select SEARCH/UPDATE by typing 4.
2. If File Name is blank, leave it blank. If it says File Name: Frontier, that's okay, too. You don't need a file name when you choose SEARCH/UPDATE. The program will look at the file in Drive 1, and find the file name
3. Press **CTRL-C**. (Hold down **CTRL** and press **C**.) A Retrieve Spec appears.

You have just completed the procedure for selecting the SEARCH/UPDATE function. From here on, whenever you see "select SEARCH/UPDATE from the PFS: File Function Menu," follow the procedure above.

Finding the Battle of the Alamo in the Frontier File

A blank Retrieve Spec is on the screen. (Make sure the phrase "Retrieve Spec" is on the bottom of your screen If it isn't, press **ESC** and select SEARCH/UPDATE again) Use the following procedure to see if the Battle of the Alamo is an event in this file:

1. Press the **TAB** key twice The cursor should be beside EVENT.
2. Type this exactly BATTLE OF THE ALAMO.
3. Press **CTRL-C** The form for Battle of the Alamo will soon appear In what YEAR did this event take place? _____. Notice what TYPE of event this is (Military). Notice the TOPIC (Texas Independence)

Ge ting the PFS: File Function Menu

When you finish looking at a form on your screen and you want the PFS: File Function Menu again, press **ESC**

Activities in this stage require students to plan a research study, identify sources of information, gather the data, check the accuracy, enter the data, rearrange the data, and eventually use the data to come to some conclusion.

Stage three involves the students initially designing the data file. We give students a research question or problem. The students have to decide what data needs to be collected, design a data collection form, develop a data file format, locate the information, and create the data file. Subject matter which is able to be organized into a data file is used to develop the questions and problems. A fifth grade health and science unit serves as a good example. What are the major organs of the body? What is the function of each organ? Where are the organs located? What is the relationship of each organ to one another? Students then design a data file format, locate a source for the information, and then collect the information.

Throughout all three stages, the students are actively involved in the learning process. Students are involved in planning and development, collecting and entering, analysis and synthesis, utilization, and answering questions.

We require the students to first identify the problems and questions that need to be solved. The problems have to be analyzed and broken into smaller, more manageable sub-problems. The information needed to solve the problems must be defined, classifed, and categorized into relevant categories. Students develop a strategy for solving the problem and locating the relevant information. Then they define a data file format and locate the sources of information.

Next, students collect the data and information and enter it into the database file. As the data is being collected, its validity and accuracy are checked. Students have to determine what information is appropriate; they have to decide which information to include and which to exclude. They have to be able to categorize the information according to the field categories. We find these student activities help to develop planning skills.

After they collect data and enter it into the database, students manipulate the information in order to answer specific questions and solve the assigned problems. Students have to understand the information they have collected. As a part of the learning process they are better able to comprehend the different relationships that may exist between the pieces of information collected.

Students interpret the data and develop answers. Reports have to be designed and printed out. Graphs and charts can be developed. At this stage, the students ... actually using the information contained in the data file. New problems and questions can be generated and solved. Additional information may have to be collected; the existing data may have to be rearranged and reorganized. At the completion of these activities, the students are better able to identify if they have solved their problems, or if they have to do more research.

Throughout this process, students are developing inquiry and questioning skills. They are learning to develop hypotheses and conduct research. As they manipulate data, they discover commonalities and differences among the information. In general, they are learning about relationships. These include: causal, geographical, historical, and spatial. In essence, they are actively involved with the data and information they have collected.

As the students progress through the stages of participation and utilize the different skills, they are becoming better learners. They are learning how to control, manipulate, and deal with vast amounts of knowledge and information. They learn how to make sense out of the individual pieces of information and knowledge they have had to deal with for so many years. By using these database activities, teachers are better able to structure the learning activities for their classes. No longer does the educational process have to include disjointed and unrelated pieces of information being spoon-fed to students.

The Toledo Public Schools use three types of database files. We purchase commercial data files; we locate activities outlined in various magazines and computer books; and we create our own activities from the curriculum.

We have used some of the Scholastic packages developed for use with the **Personal Filing System** database program **(PFS-File, PFS-Report)**. For instance, we use the **Expanding American Frontier** file, which contains 150 key events and people since 1700; **Inventions and Technology** file, which contains 130 important inventions and inventors; and the **Twentieth-Century America** file, which contains a statistical survey of the United States during the period 1900 to 1982.

All of these files can be expanded and additional records created. There are worksheets and questions included for the students. In order to answer the questions, the students have to manipulate the files with the **PFS-File** and **PFS-Report** database program. Some questions include: How did America acquire its land? Was it purchased from other countries? Did Americans fight to win control of the land? What inventions have made it possible for people to travel further distances in shorter times than they could have before the invention? These commercial data files include extensive documentation with worksheets and activity guides for students and teachers.

In addition, there are three data file formats for the students to locate information and answer questions about. These include the **American Presidents, Fun With American History** and **Local History.**

This was the first complete data file released by Scholastic, Inc. They have since released additional files including: **World Geography, Cultures and Economics; U.S. Government; Physical Science; Life Science; Poetry and Mythology;** and **Literature and Composition.**

The **World History** database from Macro Systems, Inc. has been developed for special education high school students. The activities are correlated with **Our Common Heritage** textbook. This program includes a database program developed by Intentional Educations, Inc. Everything is included on one disk and the students are taught how to manipulate the information and answer the questions by using the database program and the data files.

There are three data files included: **Family Profiles,** a collection of profiles representing a cross-section of people from different times and in different economic, social, and cultural settings; **Discoveries and Inventions,** from the domestication of animals in 9000 B.C. to 20th century technology; and **World Populations** from the year 400 B.C. to a projection for the year 2000. There is an extensive teachers' guide with worksheets and activity outlines.

We also take advantage of other database materials and articles produced by various educators, as well as those developed by our own teachers. Magazine articles have increased, as well as a number of chapters in various computer books which describe how to use database programs with students. In the reference section, there is a listing of some good resources we have discovered for database activities. We use many of these as instructional activities; they include data collection worksheets, data file formats, and a number of references for locating the required information.

The largest single source of ideas for database instruction is our own textbooks, curriculum guides, and courses of study. We look at what we are teaching and identify those areas that could be taught through a database. In order to develop database activities, our teachers have to re-evaluate what they are teaching and how they are teaching. We must look at the specific content and the ability levels of our students and determine what could be taught differently. What topics and activities could be organized into a database?

This past summer, thirty teachers participated in a co-sponsored University of Toledo, Toledo Public Schools, and Apple Teacher Scholarship Foundation Data Base Class. All of the participants had to develop three projects for use in their classes. Project 1 was to develop a complete database for use by students. The database had to include at least 25 records, with written documentation including: instructional objectives, problems and questions for student use, a list of references and sources for the data. In addition, many teachers developed worksheets, as well as printed reports.

Project 2 was to develop a database format for use by students. The format was designed by the teachers and included instructional objectives, specific problems and questions for student use. Also included was a listing of references and sources for students to use. The third project was to develop a series of questions and problems related to a specific unit that students could use a database with.

As a result of the summer program, the following activities were developed in science to teach classification, categorization, listing of descriptions and characteristics, functions, and relationships: Organs in the Human Body (fifth grade), Planets in the Solar System (special education grades four, five and six), Rocks and Minerals (seventh and eighth grade), Volcanoes (seventh and eighth grade), Earth's Geological Time Scale (seventh and eighth grade).

Figure 3
Volcanoes Data File Format

NAME ______________
DATE ______ Hr. ______

VOLCANOES

NAME OF VOLCANO:

LOCATION (Country or State):

CONTINENT: HEIGHT (in feet):

Press CTRL-N for page 2

VOLCANO TYPE: ACTIVE; DORMANT; EXTINCT

NAME OF BODY OF WATER NEARBY:

NAME OF MAJOR CITY OR CITIES NEARBY:

REFERENCES FOR INFORMATION:

Press CTRL-N for page 3
Press CTRL-P for page 1

Created by Joanne Appeddu, Media Specialist, Bowsher High School, Toledo, Ohio

Figure 4
Instructional Activities: VOLCANOES

NAME ____________
DATE _____ Hr. _____

VOLCANOES AROUND THE WORLD

Use the "Volcano Data Base" to answer these questions:

1. List the names of the shield volcanoes:
2. Name the volcanoes which are located on the European Continent:
3. What volcanoes are located near the Pacific Ocean?
4. How many active volcanoes does the U.S. have? Name them:
5. Which volcanoes are near major cities?
6. These volcanoes are composite volcanoes:
7. Name the volcanoes which are still active:
8. What volcanoes are over 8,000 feet in height?
9. These volcanoes are located on the North American Continent?
10. The extinct volcanoes are:
11. Name the volcanoes that are lcoated on the Asian Continent:
12. Does Antarctic have any volcanoes? Why?
13. These volcanoes are located near the Mediterranean Sea:
14. Does the South American Continent have any active volcanoes?
    Name them:

Created by Joanne Appeddu, Media Specialist, Bowsher High School, Toledo, Ohio

History, social studies and geography are excellent sources of information for use with database programs. In our schools, history is expanded to include scientific history, music history, art history, mathematical history and language history. Almost any curriculum area could be included. The only requirement is that the information being collected can be categorized and arranged according to some logical sequence. We have developed database activities including: U.S. Presidents, Black History, and Countries of the World.

Language arts activities include the development of a vocabulary data file. In the file are definitions, origins, spelling, parts of speech, usage or any information we want the students to learn about. In reading courses, book reports and book lists are compiled by the students and shared with others.

Data files are applied to industrial arts, specialized trade and industry classes. The files include inventories and descriptions of equipment and tools that are being used by students. Parts of an engine are also included, listing the part name, description, function, and components. In home economics, a food and nutrients database was developed, as well as fibers and child care.

Figure 5
Data File Format
FOODS AND THEIR NUTRITIVE VALUES:

FOOD NAME: FOOD GROUP:

SERVING SIZE: CALORIES PER SERVING:

MAJOR NUTRIENTS IN THIS FOOD:

OTHER NUTRIENTS IN THIS FOOD:

[SPECIAL NOTE: VALUES FOR PROTEIN, FAT, CARBOHYDRATE ARE IN GRAMS;
VALUES FOR CALCIUM, PHOSPHORUS, IRON, VITAMIN C AND THE B VITAMINS

PRESS CNTL-N FOR NEXT PAGE

ARE LISTED IN MILLIGRAMS;
VALUES FOR VITAMIN A ARE LISTED IN INTERNATIONAL UNITS (I.U.)]

AMOUNTS OF PROTEIN: CARBOHYDRATE:

FAT:

AMOUNTS OF THE FAT-SOLUBLE VITAMINS:

VITAMIN A: VITAMIN D:

VITAMIN E: VITAMIN K:

AMOUNTS OF THE WATER-SOLUBLE VITAMINS:

VITAMIN C (ASCORBIC ACID):

AMOUNTS OF THE B-VITAMINS:

NIACIN: THIAMIN:

RIBOFLAVIN:

AMOUNTS OF MINERALS:

CALCIUM: IRON:

OTHER SOURCES OF INFORMATION:

PRESS CNTL-P FOR PREVIOUS PAGE

Created by Karen Centers, Home Economics Teacher, Jones Junior High School, Toledo, Ohio

This is not an exhaustive listing; the purpose is to start you thinking about specific curriculum areas. Specific applications are dependent on the curriculum area, ability level, and grade of the students. The starting point must be what it is that you as the teacher are responsible for teaching. Determine what activities in your classroom are best suited to a database application.

First, define the topic, locate sources and references that can be used to gather information. Next, develop a series of questions for students to answer. Then, answer the questions yourself, by developing a database format and by gathering the data. As you gather the data, identify the processes students will be going through. As this process continues, refine your original questions, and possibly the file format. When you have answered all of the questions, assign the data file activities. Students can use the files that you have created, or they can create their own.

We have found in our experiences with developing database activities for the classroom that it was easy to expand and build larger and more relevant files after the initial start-up period. Naturally, our instructional activities become refined, and more activities are developed as use dictates. As our students create files, we collect them, and add them to our original files. We encourage the students to expand on our original questions and designs and add more ideas to our problems and data collection procedures.

The entire process becomes a learning experience for both teachers and students. We are able to learn more about our students' abilities and capabilities as they participate with the database activities. Both teachers and students continually update and revise plans and original designs. We try to be flexible and provide an atmosphere of cooperation, and help to create an interactive learning environment for students. Ultimately this new instructional process helps our stu- s to improve and expand their intellectual functions.

Figure 6

Student Worksheet: FOODS

NAME ____________
HOUR ____ DATE ____

QUESTIONS FOR USE WITH "FOODS" DATABASE

1. SEARCH THE DATABASE AND FIND 5 FOODS FOUND IN THE GRAIN GROUP. LIST THEM HERE. ____________________

____________________ ____________________

____________________ ____________________

2. SEARCH THE DATA BASE AND FROM THE MEAT GROUP, LIST THE FOODS WHICH CONTAIN 20 OR MORE GRAMS OF PROTEIN PER SERVING. ____________________

____

____

3. LIST THE NAMES OF ANY FOODS YOU FIND IN THE DATABASE FOR WHICH NO NUTRIENTS ARE FOUND IN MAJOR AMOUNTS. ALSO LIST WHETHER OR NOT OTHER NUTRIENTS ARE CONTAINED IN THE FOOD IN ANY SIGNIFICANT AMOUNT. ______

____

____

4. IN WHICH FOOD GROUP ARE FOODS CONTAINING MORE THAN 2 MILLIGRAMS OF IRON FOUND? LIST THE FOOD, ITS GROUP AND THE AMOUNT OF IRON CONTAINED. ____________________

____

____

Created by Karen Centers, Home Economic Teacher, Jones Junior High School, Toledo Ohio.

The first step is for the teacher to learn how to use a computerized database and to begin integrating this revolutionary tool into the curriculum with students. The ultimate result for us has been enhanced learning for students and teachers.

Howard Moskowitz is Director of Computer Education for the Toledo Public Schools, as well as part time instructor in Computer Education for the University of Toledo. Dr. Moskowitz has been involved with computers in schools since the 1960s. Beyond the regular classroom, he has run a private camp and school for children, parents, and teachers. For further information on his use of instructional activities using a database program, contact:
Dr. Howard Moskowitz
2824 Quail Run Drive
Toledo, OH 43615

REFERENCES

Baker, Patti R. (1983, September). A Software Filing System for Elementary Schools. **Educational Computer Magazine**. pp. 46–50, 91.

Dunfey, Joan. (1984, November). Using a Database in an English Classroom. **The Computing Teacher**, pp. 26–27.

Friar, Audrey, & Rotenberg, Lesli. (1985, January). Munch a Healthful Lunch!. **Teaching and Computers**, pp. 13–19.

Hopping, Lorraine, & Bollinger, Richard. (1985, April). Star Data Base. **Teaching and Computers**, pp. 15–17.

Hunter, Beverly. (1983). My Students Use Computers. **Learning Activites for Computer Literacy**. Reston: Reston Publishing.

———. (1983, October). Powerful Tools For Your Social Studies Classroom. **Classroom Computer** Learning, pp. 54–57.

———. (1984). **Personal Computers and Social Education, A Resource Guide**. Cupertino: Apple Computer.

———. (1985, May). Problem Solving with Data Bases. **The Computing Teacher**, pp. 20–27.

———. (1985, November). American Indian Data File. **Teaching and Computers**, pp. 10–15.

Hunter, Beverly, & Furlong, Mary. (1985). **U.S. History Data Bases for pfs:file.** New York: Scholastic Inc.

Macro Systems Inc. (1985). **Teacher's Guide to Macro's World History Database**. Draft Copy.

Martorella, Peter H. (1984, September). Finding Winners: Using Data Bases to Analyze Voter Data. **Electronic Learning**, pg. 58.

Naisbitt, John. (1982). **Megatrends, Ten New Directions Transforming Our Lives**. New York: Warner Books.

Office of Technology Assessment. (1982) **Informational Technology and Its Impact on American Education**. Washington D.C.: Author.

Pon, Kathy. (1984, November). Databasing in the Elementary (and Secondary) Classroom. **The Computing Teacher**, pp. 28–30.

Swett, Shiela. (1985, May–June). All-American Data Files. **Teaching and Computers**, pp. 18–24.

Woldman, Evelyn, & Kalowski, Phyllis. (1985, February). Make Your Own Dinosaur Data Base. **Teaching and Computers**, pp. 14–19.

DATABASES THROUGHOUT AND BEYOND A JUNIOR HIGH CLASSROOM

by Roxanne Baxter Mendrinos

Editors' note: Starting from the premise that the ability to find, sort through, and use information will prove crucial across virtually all sectors of future societies, Mendrinos and her colleagues have introduced database management programs thoughout their junior high school's academic curriculum. Access by telephone to a national on-line database gives students a lively connection with the world beyond the school, lending their research and report writing some of the flavor of a modern business or newsroom. Commercial publishers' databases on disk, many keyed to standard textbooks, provide more depth in specific subject areas. Teachers and students also construct and maintain their own databases; in the process they learn about ordering and accessing information as well as mastering current facts in subject matter areas.

Database searching, retrieval, and design are tools for the instructor to provide motivation, to develop deductive reasoning, and critical thinking, and to reinforce concepts and information within a particular discipline. The use of databases within the administration and in the classroom at the Edmund W. Thurston Junior High School has been very successful.

Edmund W. Thurston Junior High (EWT) in Westwood, Massachusetts, has integrated the use of databases and on-line searching strategies within the academic curricula. Electronic files of information are accessed and searched, and in some cases created, by teachers, students, and administrators.

A database is a file of records on a specific topic. Each part of the record is called a field. Data is organized into fields or categories of information. A major benefit of using a database is that information can be accessed, retrieved, and printed from a specific field.

Three types of databases are frequently used at our school. On-line databases accessed through telephone supply faculty, students, and administration with current information. **Compuserve**, one of the big three commercial database vendors in the United States, was selected by the library media specialist for on-line searching and information retrieval. Access to Grolier's electronic **Academic American Encyclopedia**, which is updated four times a year, filled an important gap in retrieving up-to-date information in areas such as astronomy or economics or biographical data on world leaders. Among many other features,

Compuserve includes UPI and AP news updated hourly, and access to the ERIC database and the Science Forum. Information retrieved is readily understood by the junior high audience.

Accessing **Compuserve** or any database vendor is easy. First, in order to access on-line databases for information, one must have a modem. We use the **Hayes Micromodem** kit for the Apple IIe. It includes an internal modem which fits into the computer; it also includes software which activates the modem. The modem changes the binary digits which transfer information from the computer into modular waves which travel over the telephone lines. Secondly, an outside or direct telephone line is a necessity to eliminate interference and ensure proper telephone reception. **Ascii Express** and **Data Capture** are two software programs that activate the modem. They include a buffer to save information on the disk which can be printed and edited at a later date.

Searching on **Compuserve** is menu-driven, which indicates certain options available to the user and allows the user to select the information needed by typing in the representative number. One can also move to different databases using the GO commands. GO AAE, for example, means to go to the **Academic American Electronic Encyclopedia.** GO NEWS gets one to the latest UPI or AP news update.

A search, primarily by subject, takes two to four minutes and costs approximately $1.00 during weekday prir . time hours. **Compuserve** is relatively inexpensive compared to the more a demic vendors, **Bibliographic Retrieval Services** (BRS) and **DIALOG** and osts $12.50 per hour prime time and $6.00 per hour after six in the evening.

A second type of database for various academic disciplines is available on microcomputer disk. Scholastic, in cooperation with **PFS:File** (Software Publishing Corporation), has developed databases on U.S. history and the physical and life sciences. Grolier has produced a database of great books and one with information on countries of the world. D.C. Heath has created databases that are compatible with and enhance their social studies and language arts textbooks. The EWT staff plans to incorporate these databases into their respective curricula. The science teacher plans to use the **PFS: Physical Science** database for searching and retrieving information on specific topics. The **Countries of the World** database will be used to access information on population density and to provide challenging and super challenging questions that will assist the teacher in thought-provoking social studies exercises. The **Great Books** database can be searched to suggest books for students who want to select outstanding examples of a specific genre. Students also can add their own book reviews to the base.

In social studies, one main problem in the study of world leaders was getting thorough, current biographical information. The situation was accented with the advent of Chernenko as Andropov's successor to prominence in the Soviet Union. The need for more powerful tools became evident. Therefore, the seventh-grade social studies team designed a database entitled "Names in the News." This database, updated four times a year in addition to print and nonprint sources of information, continues to be a valuable asset. Once accessed the information can be stored, updated, accessed, and retrieved on a database for students having difficulty locating the information in the other research sources available to them in the Library Media Center. As shown below, the "Names in the News" file includes a description of the FIELDS, the goals and objectives, and its application and use in a classroom setting.

DATABASE FILE NAME: NAMES IN THE NEWS

Objectives:

1. To establish a source of information about national and international leaders which is:
   a. Easily accessible to students
   b. Serves as a check on information obtained from print sources.
2. To familiarize students with use of a database to:
   a. Access information
   b. Add information
   c. Update information
3. To learn significant facts about national and world leaders and why they are in the news.

Database Fields:

NAME: Mitterand, Francois
SOURCE: Who's Who in the World
WHERE BORN: Jarnac, France
WHEN: 1916
EDUCATION: University of Paris (Degrees in Law, Political Science, Humanities)
JOBS IN GOVERNMENT: National Assembly, Minister of Interior, Justice, socialist Party Leader, President
JOBS OUTSIDE OF GOVERNMENT: Lawyer, Author
PERSONAL FACTS: Prisoner of War in WWII, Active in French Resistance against Nazis
RECENT NEWS: Go between for Reagan and Chernenko, wants U.S. and Soviets to resume talks

Application and Use:

The teacher and the Library Media Specialist explained the mechanics of using the **PFS File** in adding and retrieving information using the large video monitor to several slower learning seventh-grade social studies classes. Students researched their information on World Leaders using the following print sources: **Who's Who in the World, Clement's Encyclopedia on World Governments**, several encyclopedias, encyclopedia annuals, and the **Reader's Guide to Periodical Literature** for recent magazine articles. The first students to complete their world leaders' research were selected to add their information to the database. Students who had difficulty completing the assigned sheets could, after utilizing all the print sources, search the database for more information.

The students' excitement, enthusiasm, their desire to find the information quickly using print sources and to then go to the database were certainly positive (but unexpected) results. Students use their free time, study halls, and after-school study time to access the database and printed resources for information on world leaders.

In addition to the social studies database, students have since designed databases on prehistoric man and early civilizations and have created a bibliographic database on books with the most comprehensive information on each subject area researched. This will be of exceptional value next year as the need arises to quickly identify the best reference sources for each topic of study.

Databases with other classroom applications include information about taxonomic groupings in life science; data, diagrams and mythological lore on constellations for earth science; and a book review digest for the reading specialist.

A third type of database is one that can be created to fulfill a specific curriculum need or administrative task. **AppleWorks,** an integrated software program which includes a database, word processor, and spreadsheet; **Friendly Flyer**, an elementary database program; and **PFS: File** are all examples of microcomputer software programs that allow you to design a database unique to your need. **PFS: File** was choosen by EWT for curriculum applications because of its ease of use and 40-column display, which is highly visible to a class utilizing a 25-inch video monitor and computer keyboard. It is menu-driven and user friendly which means it presents one with a series of options and one only need to type the number of the option requested. Both **PFS: File** and **AppleWorks** are used for library and administrative tasks at EWT as well.

Applications of the **PFS: File** are employed in the library, guidance department, for administrative tasks. The library media specialist had designed a microcomputer software catalog. The guidance department has created personnel files for each student. The principal has created a database to store information about all the audiovisual equipment and computer hardware owned by EWT, as well as a database that lists each student and his or her respective scores in the national testing programs. This data will assist in scheduling academic students and will provide assistance in parent conferences involving levels of performance.

At the outset of our database project at EWT, staff needs fell into two areas. Teachers and administrators wanted to develop skills to search on-line databases, to create their own database, and to become familiar with classroom applications that would enhance student academic development. The library media specialist wanted to integrate the new technology into the research skill development program applicable to all curriculum areas within the school. To fulfill these needs and to provide educators with an understanding of computers and their instructional uses, the library media specialist applied for a Commonwealth inservice grant to train both faculty and administration.

Staff involvement is crucial for integration and enhancement of computer applications within school curriculum. With this underlying goal of computer and curriculum integration, the library media specialist approached the EWT staff on how databases could help students learn, improve deductive reasoning abilities, and develop research skills in each respective curriculum.

A consultant was hired through the Merrimack Education Center in Chelmsford, Massachusetts. Participation was limited to a first-come, first-served basis. Five teachers from the academic disciplines of social studies, science, and reading; the guidance counselor; the library media specialist; and the principal participated. The principal's active involvement fostered a supportive, creative atmosphere within the school environment. Five two and one-half hour sessions every other week, involving a series of lectures, small group discussions, and hands-on training, were scheduled. Ten computers were available in a lab setting in the Library Media Center in addition to a 25-inch high resolution monitor, micromodem, and access to several on-line databases and a microcomputer database program.

The staff identified the information they wanted their students to acquire, manipulate, distribute, and process. Some used categories from paper forms as fields for each database record. The participants enthusiastically developed their databases for classroom and curriculum applications or for efficient processing of administrative tasks.

The benefits at EWT of using databases in the classroom and developing on-line ırching strategies have been greater than was hoped for, and they continue to multiply with use, imagination, and creativity. Specific skills were gained in organizing and outlining information. Students were motivated to use all research materials quickly and efficiently, making use of indexes, key words, cross-references and accurate spelling for retrieval, and in identifying main ideas to meet the space limitations imposed in creating databases. Students are fascinated by on-line searching as a quick and efficient way of accessing information.

On-line searching strategies are incorporated into the seventh and eighth-grade social studies program, and in the math and academic development classes as part of the research skill program. Students receive instruction on how to access information by subject, title, and author and by cross-references using the card catalog, the almanac, the **Reader's Guide to Periodical Literature**, encyclopedia indexes, the databases designed by students and teachers, and the on-line database, the electronic **Academic American Encyclopedia.** Instruction takes place in the Library Media Center with the library media specialist working closely with the classroom teacher. Current information on such subjects as archaeology, Mondale, Gorbachev, Quebec, Sudan, Saturn, and Lech Walesa are a few of the topics searched. Students have discovered that one single resource, be it the on-line database or the card catalog, is not sufficient to answer all their information needs. They have learned to use the tracings at the bottom of a card catalog card for clues to more information and the value of indexes for efficient information retrieval.

The students successfully use the **PFS: File** database program to acquire, manipulate, process, and distribute information. The use of databases in the classroom makes students condense information in a concise, comprehensive manner so that it may be placed within the limited area devoted to one field. Thought processes and deductive reasoning are stimulated in obtaining access by using the proper subject or term. The student must decide on the main idea or the most important fact to include. Skills in reading and writing are necessary. Spelling is critical; if the student cannot spell correctly, the information cannot be easily retrieved.

The key to this exciting learning experience is motivation. Students like to interact with the computer. The lowest-level student has gained confidence in researching and adding to this database, this encyclopedia of current knowledge.

The staff at the Edmund W. Thurston Junior High School is preparing students to meet the information challenge of tomorrow as "Success in the future will more than ever before, be based on the ability to acquire, manipulate, process and distribute information. . . .The ability to use the computer will become a major tool of people who will acquire and use information." (Willis, 1984)

Roxanne Baxter Mendrinos is Director of Library Media and Computer Applications at the Edmund W. Thurston Junior High School in Westwood, Massachusetts. Experiences with computers and on-line databases during graduate school have encouraged her to develop many classroom activities. The philosophy of her school is to integrate the computer as a tool within the curriculum. Her upcoming projects include several applications of **AppleWorks** *and satellite telecommunications. For further information on her Use of databases, contact:*

Roxanne Baxter Mendrinos
Edmund W. Thurston Junior High School
850 High Street
Westwood, MA 02090

REFERENCES:

Willis, J. Spring 1984. Educational Computing: Current Status and Future Directions. **Computers in the Schools.** 1(1), 3-12.

BIBLIOGRAPHY:

An Informational Wonderland. February, 1984. **Classroom Computer Learning.**

Bring the World to Your School. November/December 1984. **Electronic Learning.**

Butcher, Diane. November/December 1984. On-line searching: How to get instant information. **Electronic Learning.**

Hunter, Beverly. 1983. "My Students use Computers: Learning Activities for Computer Literacy." Reston, VA.: Reston Publishing Company. Inside Information. July 1984. Eastern Review.

Shea, George November/December 1984. Information Services: The New Frontier of Communication. **Electronic Learning.**

The Do's and Don'ts of a Good Form Design. February 1984. **PFS: News** 1 (3).

IMMIGRANT: A Social Studies Simulation for AppleWorks

by Donald M. Morrison and Joseph Walters

Editors' note: Hortatory articles about using database management programs appear with increasing frequency in publications on computers in education, but Morrison, Walters, and their colleagues at the Harvard-based Educational Technology Center were hard pressed to find actual examples of such use in Boston-area schools. Most of the use they did find occurred in computer literacy courses, where the emphasis was on learning to operate such programs—or at best, on the concepts underlying them—rather than on the use of database management programs as tools for inquiry or exploration in subject areas. Several members of the research group responded by developing The Immigrant Unit, a curriculum unit employing an integrated applications program (spreadsheet and word processor as well as database manager) to address questions about Irish immigration to Boston during the mid-1850's. In this article, Morrison and Walters describe their group's experiences in developing the unit and in training teachers to use it.

Introduction

In 1983 the Educational Technology Center (ETC) at the Harvard Graduate School of Education was created by the National Institute of Education to develop ways of using information technologies to improve instruction in science, mathematics, and computing. One ETC research group subsequently became involved in the development of a model curriculum unit designed to explore the use of commercial applications software in the schools.

The group produced a social studies unit in which students use the integrated software package **AppleWorks** to explore the Irish immigrant experience in Boston from 1840-1860. In this unit, called **Immigrant**, students work directly with the data—actual historical records—and make decisions for an individual immigrant family. They explore database files, make decisions and purchases on spreadsheet templates, and record their decisions, feelings, and expectations in a journal using the word processor.

The development of **Immigrant** was the product of an ongoing collaboration between teachers and researchers. Teachers were consulted frequently for ideas and opinions, and two teachers were part of the design team itself. For their part, researchers at ETC contributed software expertise and a knowledge of the research literature. Although often demanding, this commitment to close and genuine teaming of school people and academics provided the energy, contacts, wisdom based on practical classroom experience, and creative sparking that made the development of **Immigrant** possible.

Origins of the Project: The Unfulfilled Promise of Tool Software

Enthusiastic reports on the use of microcomputers in the classroom have led to high expectations for the educational potential of commercial software programs, including data management systems, word processors, spreadsheets, and graphing utilities. It has been claimed that these programs create numerous opportunities for improving the quality of life and learning in the classroom. For example, by freeing students from the tedium of re-copying their written work, word processing programs may lead to improvements in the quality and quantity of student writing. By giving students quick and easy access to large amounts of data, data management programs may help students learn to think about the structure and use of information in new and interesting ways. Spreadsheets and graphs may provide ways of manipulating and exploring quantitative data, and so on.

With these expectations as background, ETC formed a working group to investigate the educational use of applications programs. This Applications Software Research Group conducted a series of interviews and classroom observations during the period 1984-1985 to pinpoint the educational impact of tool software in Boston-area schools. These investigations revealed only limited use of tool software. Typically, a word processing program or file manager would be taught for a few weeks, usually in support of computer literacy objectives. The class would then move on to some other area of the curriculum, with the result that students rarely had an opportunity to apply new tool software skills as an aid in a subject-related learning activity. Learning the tool, it seemed, had become an end in itself.

Resource Constraints

It was apparent from this survey that microcomputers, as tools, had gained no more than a toehold in the school curriculum and that, indeed, these tools were seldom being used in **support** of curriculum goals (for a similar finding, see Kay, 1985). Why?

One explanation is that microcomputers are still not readily available to the average classroom teacher. Machines are at best shared, either through a centralized computer lab or rolled into the classroom for a week or two at a time. Surely this is not enough. Tools for learning need to become part of the woodwork.

Nonetheless, the difficulty of getting a microcomputer into the classroom on a regular basis is not the only barrier to using tool software effectively, and perhaps not the most serious one. Even in cases where teachers do have relatively free access to computers, integration is often incomplete. One explanation is that teachers have general notions about using computers but have not articulated the specifics, the nuts-and-bolts, of that integration. That is, without well-conceptualized curriculum materials and well-developed models for their use, teachers will find integration of technology difficult and fitful.

Developing a Unit Using Integrated Tool Software

With these thoughts in mind, the Applications Group embarked on the development of an experimental curriculum unit designed to provide a model of the use of tool software. This curriculum development project would focus attention on the curriculum opportunities afforded by tool software in a classroom context; it would also measure the practicality of developing such units.

At that point in the project (autumn of 1984), the program **AppleWorks** had just come on the market. **AppleWorks** was then a unique program for the Apple IIe because it combined a data manager, word processor, and spreadsheet in a single program. In view of the relative power and flexibility of this program, especially when compared to its competitors, the group decided to build the curriculum unit around **AppleWorks**. The group chose to focus on the middle school level, (i.e., 6th, 7th, or 8th grade), because it was felt that students at that level could learn the software package easily.

A Brainstorming Session

In line with the desire to develop the unit in **collaboration** with classroom teachers, in early March 1985 the ETC design group invited 15 middle school teachers and curriculum specialists to attend a "brainstorming" session to generate topics that could be developed into specific units for the middle school curriculum. This brainstorming session produced over 50 potential ideas, including examination of threatened species, the establishment of a student center, planning African famine relief, publishing a best-selling novel for teenagers and so on.

As with any brainstorming session, not all of the ideas generated proved feasible. Some did not lend themselves to specific learning activities. With others it was not clear how the spreadsheet or database would be exploited. Still others, although potentially interesting to students, were not viewed as sufficiently connected to curriculum objectives. Finally, some of the more complex ideas demanded a level of abstraction bordering on dangerous oversimplification. The proposed famine relief unit, in which students would consider alternative means of delivering emergency and long-term food aid to a country in Africa, posed [illegible]sely this problem.

After considerable discussion, the group selected immigration to the United States as a topic for the model unit. This topic enjoyed a degree of curricular respectability, having secured a place in the social studies curriculum in most Boston-area schools. Also, we judged that quantitative data related to U.S. immigration would be relatively easy to find.

A Fortuitous Discovery

In the course of a library search for such data, the research assistant to the group discovered a book containing passenger lists for ships that had entered the Port of New York during the period of the Irish potato famine. The data provided in this book contained the seeds of an idea that was to grow rapidly into a workable curriculum unit involving the Irish immigration into Boston.

Briefly described, the New York passenger lists contained the following information on each passenger: surname, Christian name, age, sex, occupation, and a note as to whether the immigrant had survived the passage or "died at sea." Although these data were limited, one could not help but read between the lines to reconstruct real lives. For instance, in one family the mother, father, older brother, and infant son had died on the ship, leaving the remaining children alone to fend for themselves. What had happened to them? The passenger lists also showed that the overwhelming majority of immigrants had been dirt poor—most listed their occupation as "labourer," "farmer," or simply "servant."

In pondering the fate of these people, the development team hit upon a way of organizing the model unit. Using realistic data from that period, students would recreate the socioeconomic situation as it was experienced by the Irish immigrants themselves. To strengthen local interest, the setting would be Boston during the period 1840-1860.

Putting Together a Data Set

The data set the group eventually assembled included the following:

1. Passenger lists for two immigrant ships, including data on: surname, Christian name, age, sex, and occupation.
2. Housing lists, describing typical housing available during the years 1840 and 1850, including information on cost of rental, location and number of rooms.
3. Job lists, describing representative jobs available for the **same** years, 1840 and 1850, including information on wages, location of job, and experience required.

In addition, two spreadsheet templates were constructed:

1. A "market basket"—designed to allow students to calculate a family's food and clothing expenses over a period of one month.

2. A household accounts template—designed to allow calculation of total family income and expenditures for up to a ten-year period.

Finally, the unit included a map showing Boston and surrounding communities as they appeared during that period, and a table displaying the means and costs of commuting between various neighborhoods. All of this information was gathered in the course of a few visits to area libraries, from a relatively small number of sources.

The Simulation Itself: Multivariate Decision Making

The assembled information provided the basis for a simulation in which students can "adopt" an immigrant family and then explore the world of the early Boston Irish from the perspective of this adopted family. In the simulation students decide where to live, what jobs to take, how many family members to put to work, and what food to eat.

These variables interact in interesting and complex ways. For example, better paying jobs are associated with higher transportation costs. A family may attempt to find housing nearer these better jobs, but such housing tends to be more expensive. This in turn offsets any potential increase in income. Similarly, having more children means having more mouths to feed, but it may also lead to an increase in the number of breadwinners. At the same time, if children are put to work too early, their future earning power may be seriously limited by their lack of education.

The computer assists, but does not direct, the decision-making process. Information about jobs, housing, and transportation costs is contained in database files. Food costs and overall family budget are calculated using spreadsheet templates. Each group of students keeps a family diary, a running record of the group's decisions, using the word processor.

Because the data constitute a reasonably accurate representation of the historical situation, the collective decisions made by the different student "families" during the course of the simulation tend to reflect the experiences of the actual Boston Irish immigrants. For example, it is an historical fact that economic constraints forced most of the Irish to live in tenement housing near the city center, to work long hours at low-paying jobs, and to send their children out to work at an early age. **Immigrant** thus successfully models a specific historical situation, the immigrant experience in general, the cycle of poverty, and the dynamics of urban geography. Significantly, this information comes not from a textbook summary, but from an active in-depth analysis of the situation from a micro-level perspective.

Piloting the Unit: A Workshop for Teachers

To extend the collaborative effort and provide further classroom expertise, the group issued a general invitation to area junior high school teachers to attend a summer workshop. The workshop spanned three 6-hour days, simulating (to a degree) a junior-high social studies class. The workshop began with a brief introduction to the problem, some discussion of immigration, and a reading of a story about a typical immigrant family. The teachers, taking the role of students, divided themselves into teams and set to work selecting families, finding housing and jobs, working out food and clothing budgets, and then using the spreadsheet template to keep track of overall income and expenditures.

As planned, the groups periodically produced diary entries in which they recorded their family's economic experiences and decisions. The diary entries were posted on the class bulletin board, along with a large map with flags showing where each family had chosen to settle. In this way the participants were able to keep track of the progress of the simulation at a "macro" level.

The workshop focused directly on the simulation itself and provided little direct instruction in **AppleWorks**. Although some teachers were bothered by the lack of explicit software instruction, most appeared able to master the necessary skills with relative ease. More importantly, the materials proved engaging to these adults at several different levels. Some groups, for example, became caught up in a moral dilemma about whether to send younger children out to work to bolster the family income. Another group of male teachers, having adopted a single man to follow through the simulation, first found a wealthy spinster and married her!

In keeping with the collaborative spirit of the development project, teachers in the workshop were urged to provide suggestions for improving **Immigrant**. Some of these suggestions were incorporated in later drafts of the materials. For instance, several teachers felt the need for an element of chance and risk-taking. Others felt the need to consider the larger community of immigrants, especially in terms of how families communicated and looked out for one another. Many of these suggestions were written into the Teacher's Manual as suggested extensions of the unit.

The story of the development of **Immigrant** is not over. The materials are now in the public domain and are being disseminated by ETC on an **AppleWorks** disk along with a teacher's guide. Reports of teachers trying the materials are already starting to come in. For example, two of the teachers who attended the summer workshop are now planning a classroom pilot of **Immigrant**, scheduled for November, 1986. They plan to augment the computer-based materials described above with a slide show, a field trip to Boston's North End, and a talk by a local historian. In view of the ease with which the materials can be extended and modified to meet local needs, it will not be surprising to hear of **Immigrant** being used in other innovative and unplanned ways.

Conclusion

Our hope is that **Immigrant** will provide teachers and curriculum developers with a reasonable model of how a sophisticated commercial software package can be exploited to meet curriculum goals in a specific subject area. In particular, we have tried to show how a program like **AppleWorks** can be used to help create an environment that allows the student to gain an insider's view of a complex system, thereby affording an opportunity to probe its constraints and hidden laws at first hand. In this case, the system happens to the world of the Irish immigrant in mid-19th century Boston, but we see no reason why a similar approach could not be used to explore other worlds in other subject areas.

The research and development described in this paper were supported in part by the National Institute of Education (Contract # NIE 400-83-0041). The curriculum unit **Immigrant** was jointly created by Jane Manzelli, Henry Olds, Debbie Ross, and the two authors. Opinions expressed herein are not necessarily shared by NIE and do not represent Institute policy.

Both Donald Morrison and Joseph Walters have stayed close to the cutting edge of research and development on educational applications of microcomputers. Donald ("Chip") Morrison received an M.A. in Language Studies from the University of Hong Kong in 1980. He is presently a doctoral candidate in Human Development at the Harvard Graduate School of Education.

Joseph Walters received a doctorate from the Harvard Graduate School of Education in 1982. He is currently a research associate at Harvard Project Zero, where he investigates the role of microcomputers in human learning. For further information about **Immigrant**, *contact:*

Donald Morrison and Joseph Walters
Educational Technology Center
Gutman Library
Harvard University
Cambridge, MA 02138

IMPROVING TEACHING EFFECTIVENESS THROUGH INTEGRATED SOFTWARE

by John Rouleau

Editors' note: An integrated software package made "outcomes-based education" manageable in John Rouleau's classroom, setting off a revolution in his students' orientation to learning. By defining instructional objectives precisely and requiring students to demonstrate achievement of one set of objectives before moving to another, Rouleau made it impossible for students to survive in passive orbit around the teacher. Instead, they had to take responsibility for their own learning, with Rouleau serving as a resource and as a source of feedback about their progress and problems. Without an integrated software package, however, tracking, reporting on, and analyzing two classes' progress on 125 distinctive objectives would have been infeasible. With the software, Rouleau was able to do all of this and to adjust his own teaching in light of the analyses. In this case, the computer served as a tool for the teacher—an instructional management tool—and thus made possible a real shift in students' instructional experience.

You never know when some mode of reinforcement will crop up and push your teaching in new and better directions. One recent occurrence at Rock Point School was the coming together of **AppleWorks** and outcomes-based instruction methods. Each contributed to a highly successful health program that motivated both teachers and students.

The first impetus for change became apparent last spring, when I felt that my students were leaning too heavily on me for responsibility for the classroom. They were delegating to me too much control and accountability for their education. The result was that they could sit back and be bored, entertained, successful or not, but in any case, not responsible. I was fortunate to have a teacher from another district introduce me to the Johnson City style 2 of outcomes-based education. Through the spring and summer, as I incorporated that teaching style into my plans for this year, I could sense the return of equilibrium in responsibility for the class; I was getting back to sharing with the students in their education process and getting away from the dominant and equally powerless nag I had become. I anticipated that changing from a grading system of A, B, C, D, F to A, B or Incomplete, with the onus on the student to resolve the Incomplete into an A or B, would require a lot of adjustment for both me and the students. I was pleased to discover, however, that the rationale for grades for performance rather than time served, clearly agreed upon goals, and the open architecture of an outcomes-based curriculum were sufficient to win most students' acceptance. They went from, "You mean I can't fail?" through "Let's see how this actually works at report card time," to a feeling of confidence and competence. As the first quarter's grades came out, I was congratulating myself and my students for making the transition to the new grading system so smoothly, when one student came in to complain long and bitterly. She had always gotten D's and F's; she felt she had earned a D; the Incomplete on her report card was an insult and an abomination. Unspoken in her complaint was that the Incomplete meant she couldn't mark off one more quarter of her twelve-year sentence to education and proceed to float through another quarter. In the end, she took three months to make up the Incomplete and earn her B. In the process, she became a staunch supporter of outcomes-based education.

Rock Point School is a residential school and, therefore, we found that our need to teach health-related issues was even more clear than a day school's might be. We found again and again, from discussions with day-time teachers and evening dormitory counselors, that issues of adolescent health and development needed more attention than could be given in individual sessions. Consequently, we decided to insert health-related curriculum into the regular academic curriculum. Just as the summer contained planning for an outcomes-based teaching style, it also contained planning for a health class.

That summer also brought one other factor to bear: the introduction of integrated software, specifically **AppleWorks**. We have had microcomputers in the science lab for several years, but they have been used mainly for teaching BASIC programming and then writing programs to solve particular problems. We relied mainly on our own programming skills for software, both for education and administration. **AppleWorks** is an integrated program which combines word processing, a spreadsheet, and a database in a humane and powerful way. The word processing part of the program is straightforward and easy to learn. It is not limited by a narrow range of options for controlling the printed product, or made clumsy by complicated keyboard routines to do common word processing tasks such as changing margins or editing text. Information in both the database and the spreadsheet can easily be moved into a word processing document for formatting and printing. That same information can be accessed directly in the database format, where it can be sorted, listed, or counted. The spreadsheet is like a piece of ledger paper with 256 columns, many more rows, and a waiting genie who will do any math function with any of the information on the paper. **AppleWorks** is clearly more sophisticated and useful than the programs we had written for ourselves. We began using this software both to help manage instruction and as part of the actual classroom activities.

The introduction of behavioral objectives into the curriculum was a fundamental change. We wrote curriculum objectives which contained (1) the conditions under which the student will be tested, (2) the behavior that will be tested, and (3) the criteria which will be used to judge mastery. For the health class this year, it has meant keeping track of progress of each student in two classes on about 125 objectives—a nightmare for pencil and paper techniques but a piece of cake for a machine. Basically, what the machine does is offer a chance to design looseleaf notebook (database) of about 800 identical pages. Each page contains labeled blanks for information. The work of putting information into a computer is no more time-consuming than putting it down on paper, and the advantage is that the machine can sort the information in a host of ways. For example, I was able to see how many students had mastered each objective and then plan where to take my teaching next. I published individual reports for all students that showed exactly how they were doing and thereby made them the author of their success, rather than the victim of grades. These reports are a very motivational form of feedback. They list the students' successes and make whatever work still needs to be done seem manageable. Students who used to throw up their hands at a mid-semester warning—"I'll never catch up with all I've got to do!"—now ask for progress reports. The machine sorts the current information and prints it quickly and clearly. Students then check the objectives they've mastered and make plans with me or other students to work on those objectives which need more work. Easy access to their records and the emphasis on work to be done, rather than failures to forget, has brought me and my students into a more cooperative and supportive educational style. The increased cooperation has tended to blur the distinction between correctives and enrichment activities; we are more in common cause. The benefits to me as the teacher are that I can arrange the information for grading purposes or curriculum development purposes: if 90% of the students are mastering all but two objectives in a unit, I will want to change how I teach that unit. I can also tag each objective so that I can pull out how students are doing on a particular type of objective. For example, I might want to see how they are doing on just the alcohol part of the drugs and alcohol curriculum. Our school sends home narrative as well as letter grades four times per year. With **AppleWorks**, I can merge the list of mastered objectives from the database into the health class grade written on the word processor. In sum, being able to manipulate detailed information about behavioral objectives has made my teaching more effective and has brought students much more into the arena of responsibility than before.

Part of the curriculum this year involved teaching students about money. We set up some in-house banks (The Vermont Gravel Bank, The Snow Bank of Vermont, Ocean Tide and Trust), issued checkbooks, and then had the students write checks for the apartment downtown which they had "rented," deposit paychecks for the jobs they had "applied for," and buy goods and services within the school. All this entailed massive amounts of paperwork. Just printing a bank statement for each student each month would have put an end to the whole project without the computer. The same machine that keeps real banks from disappearing under a mountain of paper enabled us to operate our banks in our spare time. Using the **AppleWorks** spreadsheet, we were able to keep track of each student's checking account, with service charges and all the deposits, canceling, withdrawals, etc. with a minimum of effort. The spreadsheet did the math, and when it came time for printing monthly statements or weekly paychecks, we merged spreadsheet information with the word processor. Most of the work was involved in setting up the program, printing the checks, and figuring out how the system would work, but that was done in the summer in the relative cool of the academic year. When the students came, we were ready and able to run a bank without burning out on paperwork.

Another part of our program which was facilitated by **AppleWorks** database was using a trivial pursuit-style game format in the classroom. The game is an ideal way to rehearse and reinforce small bits and pieces of a curriculum, for example: Name three food sources of vitamin A. Which part of the heart pumps blood to the lungs? Name three effects of cigarette smoking on others. Where should you open your mail? Questions were written by teachers and students. By putting the questions into a database, we were able to publish them in a standard format and also arrange them by content, or by behavioral objective, or by date. In other words, we were able to select out the questions we wanted to emphasize in any part of the program. If some students needed correctives for understanding the digestive system, we could select out questions that would emphasize that aspect of the unit. If we were looking for enrichment for some students in nutrition, we could select out questions for that goal. The advantage of the game format is that the game is self-sustaining; i.e., once students get the hang of playing, they become a very intentional self-motivating group within the classroom, and they free up the teacher to work with another group of students. The point is not that we couldn't have used the trivial pursuit idea without the machine, but that the machine made our work so much more widely useful and effective. Once again, most of the work was done in the summer in the planning stages of the course, and we were able to make use of the work all year long.

As this year comes to a close, we are in a position to evaluate the past year and look ahead to next year. It is clear that **AppleWorks** has earned a place in our classroom. We have every intention of using it again, in the same ways, next year. At the same time, it is equally clear that we haven't exhausted our possibilities either. Two projects on the agenda for the next school year are using database for counseling and goal-setting, and also for teaching statistical research in the classroom. The database is tailor-made for collection of data and then analysis from several viewpoints. As students become more familiar with manipulating a database, they begin to see how it can be used as a research tool. Keeping track of weekly and long-range goals on a computer is flirting with gimmickry, but, for some students, it provides motivation and a technique for developing control of their lives. We are experimenting with both database research and database counseling this year, and they look promising.

I don't make any claim that using an integrated software program will make teaching any easier or more restful. Working with young people will always be a challenge to the creative and nurturing part of us. What stands out in well-defined relief to me is how much more powerful and far-reaching my teaching has become as a result of the coming together of two elements: development of a curriculum incorporating outcomes-based education techniques and an integrated software program.

REFERENCES

AppleWorks Copyright Apple Computer Company 1983
20525 Mariani Avenue
Cupertino, California 95014
Johnson City Central School District
666 Reynolds Road
Johnson City, New York 13790
Dr. Albert Mamary, Superintendent

John Rouleau is Assistant Headmaster of Rock Point School, a private residential school in Burlington, Vermont, where he teaches natural history, biology, chemistry, health, French and photography. John was selected to be a consulting teacher in the Vermont State Department of Education Resource Agent Program for the innovative health program described in this article. For more information on using integrated software, contact:

John Rouleau
Assistant Headmaster
Rock Point School
Institute Road
Burlington, VT 05401

III OTHER APPLICATIONS THAT MAKE ROOM FOR TEACHERS

MICROCOMPUTER-BASED LABORATORIES: CONNECTING COMPUTERS AND SCIENCE

by Ted Hall

Editors' note: With the right software and peripherals, computers can bring graphs to life, Ted Hall finds. In fact, they can bring a broad array of scientific principles and processes to life. In microcomputer-based laboratories, devices ("probes") that measure temperature, heart and respiration rate, motion, light, pH, and other phenomena are used to take data which accompanying software can display and manipulate. This not only speeds the process of experimentation, but also permits students and teachers to focus on description and modeling of the phenomena under study rather than on the tedious tasks of data recording and display. In the process, they are gaining experience in using computers as tools, as do researchers in virtually every domain of modern scientific inquiry.

Never had I seen students so interested in a graph. This graph was different, however. The graph was appearing on the monitor of a computer that was connected to a temperature probe measuring temperature versus real time. This particular group of students was a general-level chemistry class for eleventh graders. None of the students in the class particularly liked math; they always had great difficulty in understanding the purpose of graphing data from their laboratory investigations. Suddenly, the computer was doing all of the work for them. Instead of taking one period for data collection and following up the next day with a day of graphing and interpretation, we were able to see the trends in temperature as they actually happened.

The concept that I was trying to teach was heat of fusion. It is very hard for less able students to understand that once a phase change from solid to liquid begins for a substance, the substance remains at the melting point until all of the solid is melted. The classic experiment for demonstrating this concept is the cooling curve of sodium thiosulfate. The classic experiment is not exactly the most exciting to observe in the lab. You take a sample of sodium thiosulfate and heat it until it melts, then allow it to cool as you record the temperatures at one or two minute intervals. Data collection continues until all the liquid has changed back into a solid. After the data collection is complete, the student then constructs a graph of the data and answers leading questions about the graph that lead to a conclusion that there is an energy change that occur s, even when there is no change in temperature while the solid is forming.

photo credit Phinney Morrison

With the temperature probe attached to the game paddle input of the Apple II computer, and a commercially available program called **Temperature Grapher**, we were able to do the whole experiment in one period, have a graph available, and answer all of the important questions as the phase change was occurring My first use of a "microcomputer-based laboratory" experiment was with one computer and all of the students gathered around a small monitor. In spite of the limitation of equipment at that time, I felt that the concept was understood by those students more completely than I ever had seen from the traditional approach to the experiment.

That first experiment was enough to convince me of the worth of microcomputer-based laboratories (MBL). Since that time, many other applications of MBL have been developed. The pioneers in this field are at the Technical Education Research Center (TERC) in Cambridge, Massachusetts. Bob Tinker, as director of TERC, has developed probes and accompanying software that measure heart rate, respiration rate, motion, pH, and light. Applicatons of MBL extend from elementary school science through graduate school, through all ability levels.

Although my own teaching has all been at the high school level, I have seen MBL used successfully at the elementary and middle school levels as well. One particular application was in a sixth grade class in Concord, Massachusetts. A motion detector has been developed using the same principle as the automatic focus mechanism on cameras. The experiment that I observed made use of the motion detector and the motion of some active sixth graders. The students could watch on the screen as they walked back and forth, varying their speed as they moved. Without trying to teach terms like velocity and acceleration, the teacher successfully presented those very concepts through the wonders of the computer. The software that accompanies the motion detector will draw a graph of distance versus time on the top of the screen while a graph of velocity versus time is being drawn on the bottom half of the screen. The students could see graphically and immediately the effects of speeding up and slowing down and of moving forward and backward. The students particularly enjoyed an activity where they were asked to move in a way that would duplicate a graph that was given to them on paper. In this way, the student made the connection between the graph and the actual physical event in reverse of what we usually try in our laboratories.

photo credit Phinney Morrison

Another exciting application that has been used in science as well as health and physical education classes is the HRM software package **Cardiovascular Fitness**. By clipping a sensor to the ear, the student can observe changes in his/her heart rate as a result of exercise, diet, or even embarassment! A student can record changes as (s)he rides on an exercise bicycle or runs in place, making observations exactly at the moments that changes occur.

As computers make their way into the classrooms, it is important to plan for their infusion into the curriculum. In science, there are many tutorial, drill and practice, and simulation programs available for teachers to examine and use in their classrooms. These applications often do not get at the essence of science instruction in the same way that MBL does. Almost all professional scientists have computer-based instrumentation in their laboratories. Science classrooms need to reflect this trend. MBL gives us new tools to help students in developing their science process skills.

At the secondary level, MBL will provide teachers with tools that will make their laboratories more useful and more efficient. Teachers who are comfortable with their science curricula can add MBL to their program without any significant change in overall objectives and topics covered. But MBL may signal a major change of focus in science education. Science courses can adopt a topical approach to scientific problems and make use of MBL in solving those problems. All this can be accomplished by clustering appropriate traditional teaching objectives around some area of interest to the student. An example of this approach might be in the area of human physiology. Students are often more interested in their own physiology rather than the physiology of some lower animal. By experimenting with human physiology, students will actually explore the topics, rather than have those same topics fed to them by eager teachers. With inexpensive light and temperature probes, students can measure skin temperature, respiration rate, heart rate, and response time. From their measurements, they will come to some conclusions about the physiology of their own bodies. And more importantly, the students can then ask the "what if..." questions and be able to answer those questions with self-designed laboratory experiments. Other topics that can be explored in the same way include music and sound, fitness and athletics, weather, food, acid rain, radiation, perception, or any other particularly interesting topic.

One of the most valuable experiences for students is the realization that they can be involved in the process of science. This can occur at the elementary level as well as at the secondary level. By allowing students to "play" with the probes that measure temperature, sound, motion, and light, teachers will see that curiosity of young children will lead to some interesting conclusions. An example of this kind of playing can be observed in a classroom using the **Bank Street Laboratory**. This package allows the students to explore sound. The activities allow for measurements of loudness and pitch, as well as the ability to record, display, analyze, and play back sounds. Students who are allowed to play with loud and soft noises in a classroom are certain to enjoy themselves, and they are bound to come to some conclusions about the amplitude of sound. Students will, when given the tools, explore their world. MBL gives students the capability of immediate feedback, allowing them to become participants in the process of science.

Is there anything wrong with MBL? MBL may sound like the savior of science education from all the praise being heaped on it. What are the drawbacks? One teacher asks, "What about graphing skills if the computer does all of the work of data collection and manipulation?" The answer to this question lies in the classic statement that the computer only does what it is told to do. Students still need to be able to understand scaling, labeling of axes, ranges of values, etc. if they are going to have the software and hardware work effectively for them. MBL provides tools for the laboratory, just as balances, test tubes, burets, and microscopes provide tools for the lab. The tools that MBL provide are more sophisticated, but they still need to be totally understood if they are to be used to their greatest potential. An obvious question is the cost of these materials. Luckily, the cost of the software packages that include the hardware for making measurements are in the affordable range ($60-$300), and for those teachers who like to build classroom equipment, directions and parts are readily available at very low cost. Another question that is bound to surface is the durability of the equipment. What happens when some solution gets spilled all over the computer?...What happens when an overactive second grader breaks the probe? The answer: accidents are unavoidable, whether you are working with MBL or the traditional lab. The replacement cost of a temperature probe is not much different from the replacement cost of a thermometer. The replacement cost of a computer is much greater, so the computer simply needs to be protected in the lab. Simply stated, the benefits of MBL far outweigh the drawbacks.

photo credit Phinney Morrison

MBL can really cause significant changes in the teaching of science. This approach to science education will stimulate the problem-solving and creative thinking skills that may have been lying dormant in students who have been simply learning scientific facts in a more traditional science program. In a microcomputer-based laboratory, students can easily become involved in the process of science. That process stimulates students to ask the important questions and allows them to investigate the answers to those questions.

Ted Hall has been teaching science for eleven years. Though he has used computers occasionally, Ted confesses that he wasn't excited about them before being introduced to microcomputer-based laboratories. Now his goals are to build a well-outfitted lab with equipment for many applications and to acquire portable data gathering equipment for field experiments. For further information about his use of MBL, contact:

Ted Hall
Wayland High School
Wayland, MA 01778

FILLING A GAP IN READING COMPREHENSION PROGRAMS

by

Rebecca B. Corwin, Joanne Bronga and Charlotte Reid

Editors' note: In this article, Corwin, Bronga, and Reid greet a new genre of software for reading instruction, one that goes beyond "skill-and-drill" to foster comprehension through a learner-centered approach characterized by learner control over learning goals and/or means of achieving them, informational as contrasted with judgmental feedback, and prediction and approximation as the learner approaches the goals. Although the authors illustrate their notions about powerful approaches to comprehension instruction principally by reference to a single piece of software, their real focus is on the approach rather than the program, and they mention several additional programs that take similar approaches. Two of the authors are teachers of high school special needs students, and examples of applications for remedial students abound, but the third author's experience suggests that the same line of approach is effective for developmental instruction with middle grades students, as well.

Based on our experience in reviewing computer software for reading instruction, we can only agree with Susan Ohanian (1984) that the "skill-and-drill fix is about as helpful to children's intellectual development as sucking on jujubes." But finally it looks as though appropriate reading comprehension software does exist. A variety of software which focuses on more than isolated decoding skills is becoming available. In this article we will focus on **GAPPER,** which integrates many skills; we will also identify other pieces of software which do other parts of the job, presenting material in a context of reading for meaning.

Of course, the computer cannot form the center of a reading program—good books ought to. We all know students who have been turned off to books, though, who are nevertheless willing to work with the computer. Many of the programs we have found point the way to software developers and give us hope that some good software will allow our discouraged students to practice reading skills in appropriate ways. Two of us, as high school special needs teachers, see good potential here for older remedial students. The third, a fifth grade teacher, sees a good tool for middle grade readers of all types.

In this article we will look at a model of powerful use of the computer in reading comprehension. That model is **GAPPER**. Other pieces of software with similar flexibility and power, which cover some of the same material, are also identified.

The major criterion for selection is, however, that the program treat reading as an exercise in meaning. Beyond that, we have added criteria suggested by Mokros and Russell (forthcoming) for learner-centered software. The software should:

1. Allow the learner control over either the goal, ways of approaching the goals, or both.
2. Provide feedback which is informational rather than judgmental.
3. Encourage prediction and approximation as the learner approaches the goal more and more closely.

We do not intend this choice of software to exclude other good software; we are simply reporting on those which we have used to date.

What is the approach?

GAPPER, in its most basic form, is a reading comprehension program which uses a variety of texts. There are two disks: the game disk and the text entry disk. The game disk sets up the five-part game procedure for the student, and the text entry disk serves as an editor, allowing the teacher or students to enter their own text selections. The game disk is the focus for the student activities. In the game, as constructed, the students:

1. Preview comprehension questions: Students have the chance to look at the comprehension questions which will be asked after the text passage is read. This selection allows students to focus their attention on the main idea and to read with a clear objective in mind.
2. Read the text: The student does a timed reading of the text passage.
3. Answer comprehension questions: A report of estimated reading speed in words per minute is then provided and the five multiple choice questions are asked about the selection.
4. Replace words deleted from the text (the "cloze" procedure): During the fourth phase of the game the reading passage is again displayed with words randomly deleted and replaced with blanks. The player can earn points by filling in the words correctly. Hints are provided for incorrect responses.
5. Predict text: In the final phase of the game, called **Star Text**, all words have been deleted from the passage and replaced with asterisks. The player earns points for each correct entry of a word from the original text.

Thus the student reads the text at least three times, previews and answers a set of questions, and, most important, develops the skills of the most powerful readers: prediction and self-correction. The cloze activities, over a familiar text, are some of the most worthwhile for developing intelligent readers.

Typically, many students enter the fourth grade with good decoding skills. They know the phonics rules and can read aloud reasonably well, but as the reading textbooks shift to an emphasis on comprehension, they become lost. An over ›hasis on phonics can create in some children the idea that reading is supposed to be perfect the first time—no guessing, no errors. That is exactly wrong. Most powerful readers invent much of what they read. They imagine scenery which isn't specifically mentioned, because it helps the story go along better. When they come to an unfamiliar word, they predict, and if they can't make sense of the prediction they simply continue to read. Like conversations, written communication can be less than completely precise; we fill in gaps and adjust to fit our own sense of the world (for background, see Smith 1978; and Holdaway 1979).

GAPPER allows students to do what they need to—to predict when words are left out, and then when the entire package is erased. During the running of **Star Text**, when the only clues are asterisks where the letters were, many students are shocked at how much of the text they can recreate. This makes them feel powerful and in control of their learning, and shows them the power of prediction and self-correction as a reading technique.

Using GAPPER with a class

There is an anthology of readings which is geared to the fourth through eighth grade levels, and each disk in the **GAPPER ANTHOLOGY** contains five selections focusing on a particular skill (e.g. spatial visualization, description of physical objects, simple definition). These selections form a good resource pool for working with the skills involved: the teacher boots up the game disk and students can use more than one computer to work. Pairs of students are encouraged to work together, either competitively or cooperatively, so this could then be used to encourage communication about reading selections—an unusual thing for many uncertain readers.

Some teachers set up the computer so that a number of related activities and materials are nearby—a kind of learning center approach. This is especially effective in a resource room or a classroom in which students move about to various centers or corners during the instructional period. We are also finding that having books of essays and bound books of other students' stories allows those who are using the computer to take a break from the program and have related materials right there to work with, browse through, or borrow.

An interesting aspect of the program, especially for intermediate grade students: as they guess the words in **Star Text**, using only the asterisks as clues, the machine beeps for each time a word appears in the text. If you guess "the" and type it in, it not only appears in the text wherever it was originally written, but you get points for each correct guess. Many passages contain the word "the" twenty or thirty times. Since guessing a frequent word generates more points for the students, they often begin to care about word frequency. One very successful group focused most of their effort on frequency analysis; since their teacher was very interested in multi-disciplinary computer use, she had them keep records on one machine, using a graphing package, while using **GAPPER's Star Text** on the other. They printed out the texts, their frequency charts, and became deeply interested in analysis of the frequency of words in English. The graphing work they had done served to connect reading and math—a much more comfortable area for them.

Other programs; extended uses of GAPPER

The huge power of the computer in a reading program lies not in game playing by itself, but in the user's ability to control the level and content of the text. Because of these capacities, the programs we mention serve as tools for both teacher and student rather than as limited-function drill programs.

Naturally, word processors ought to be included on this list. For preparing text, without a need for a shell program which contains a series of activities, they are the simplest, most sophisticated tools on the market today. We assume that their use is becoming deservedly widespread in education.

Programs we will mention which allow control of text within a shell program include the following:
MYSTERY SENTENCES, published by Scholastic. This program enables a student to guess a sentence from a series of blank spaces, getting points for correct guesses within the game format. Three clues about the meaning of the sentence are provided to help in guessing; otherwise the student can choose letter clues and can guess from the contours of the words in the sentence.

CROSSWORD MAGIC, published by Mindscape. This is a tool program which allows students to generate their own crossword puzzles. After entering words which are automatically arranged in a puzzle template, the student writes clues for the words. The completed puzzle can be printed in hard copy or played on the computer.

M-SS-NG L-NKS, published by Sunburst. As in **GAPPER's Star Text**, this program asks the student to guess the text from visual clues provided. Students may choose to have all consonants or vowels eliminated; they may choose to have alternate words removed, all but the first letters removed, or all letters hidden. Students must develop a strategy for finding the missing letters in order to complete the sentences or paragraphs.

Editing capability allows the teacher to prepare text selections which are geared to her own class, the interests of her students, and their reading and experience levels. Eventually learners can enter their own text, as well. Although the process is tedious at first, and a little clumsy, it's definitely worth it. Imagine the following (especially if you work in a resource room situation with a significant range of student reading levels):

- A severely learning disabled fourth grader works with **MYSTERY SENTENCES** in one lesson in the resource room, guessing words and getting clues. He does a reasonably sensible job of determining when to ask questions. In the next session, however, he decides to make up his own sentence and his own clues, and spends very concentrated time in determining and refining his ideas of what makes clues hard or easy. When he returns to the computer the next day, he can read what he's written for the first time ever.
- A learning disabled junior high school student writes the comprehension questions for a section of **GAPPER's** text. She must develop four reasonable responses on the multiple choice format, and decide on the feedback for the correct and incorrect responses. Although it takes her a week to work it through with a partner, she is completely confident that her "quiz" is well-constructed. She prints out copies to take back to her classroom for classmates to work on.
- A third grade student uses **CROSSWORD MAGIC** to practice his spelling words for the week. Because he enjoys creating the definitions so much, he proposes that he make a puzzle for the whole class to study their words from. He spends Monday in the resource room developing the puzzle, and the class uses it during the week.
- One sixth grade student's story, written as part of his continuing adventure series, is edited, entered, and prepared for other students to read on a **GAPPER** disk. Although the teacher had to do the typing and final preparation, the student hung over her shoulder the whole time, watching the process and kibitzing. He wrote the comprehension questions for the story, and was surprised that there was so much work to the creation of multiple choice questions.
- A fourth grader decides to enter a part of **The Lion, The Witch, and the Wardrobe**, the book her classroom teacher is reading out loud. She works to choose a good, self-contained piece of it, and then carefully decides which words should be deleted on **GAPPER's** cloze exercise. When it's finished, she puts the disk in the classroom library for others to work on. Next week she plans to enter it on the **M-SS-NG L-NKS** disk in her homeroom so that it is accessible there as well.
- A second grade teacher enters text into **GAPPER** which is selected by the children from their vast store of rhymes, chants and folk tales. The familiar words allow children to be very successful at prediction; two of her "non-readers" work with the computer over and over, using the same text.
- A seventh grade English class writes and edits science-oriented compositions to put into **GAPPER's** game disk. Their work is printed out on the word processor (**GAPPER** is relatively easy to use) and shared. At the end of a week, the work is evaluated by the class as to the kinds of work they

want included in their anthology. An editorial board develops comprehension questions onto disks. They donate the disk and the printed-out copy (called "Readings in Science") to the school media center.

Older remedial students

Secondary special educators spend a great deal of time in teaching reading in the content areas, with a lot of emphasis on finding main ideas and sorting out the supporting details. Often, as well, there is emphasis on sorting out inference from fact, and proving that in the context of reading selection. A reading comprehension program with room for teachers and students to enter text allows students to work with resource room teachers on precisely the skills required for the texts they are using in sociology, history, or other subjects.

With a modicum of cooperation from subject matter teachers, the resource teacher can determine the critical chapters and passages from textbooks, develop questions, and prepare subject area-oriented text for **GAPPER**, **M-SS-NG L-NKS**, **MYSTERY SENTENCES**, or **CROSSWORD MAGIC** for students. Over time, these home-made activities can stand other students in good stead, allowing them to choose practice exercises easily and quickly. Particular skills can be emphasized and the needed vocabulary highlighted.

Other uses with older students

- A seventeen-year-old interested in rock stars enters the text of **People** magazine's most recent interview with Prince into **GAPPER**. He creates comprehension questions, participates in the other activities, and stays with the task for long periods of time.
- A fifteen-year-old interested in sports cars dictates a story to the teacher which is entered first into **GAPPER** as text. Next, she edits it into a series of **MYSTERY SENTENCES** with supporting clues. She finally donates it and a related **CROSSWORD MAGIC** to the resource room teacher in the elementary school next door.
- A teacher with a number of very low-reading students finds texts she feels may be appropriate in terms of her students' interests. She enters the text and **GAPPER** analyzes the grade level for her, so she does not need to guess or go through tedious manual analysis. The Final Preparation process includes text analysis which gives information about the number of words and sentences in the text, the average sentence length and an estimate of the average number of syllables per word. This information is used to compute a readability index (the Flesch index) and an estimate of the grade level the text is appropriate for.
- A special needs student brings the weekly vocabulary words from mainstream classes to the resource room. Using **CROSSWORD MAGIC**, she generates clues for each word and uses her puzzles as a study guide for vocabulary tests.

Special needs work

This kind of comprehensive and flexible software offers a great deal to the special needs teacher and student. First, it is easy to use. Once the text is entered, the students are completely independent of the teacher. Most of the work (except for entering text and missing words) is single-key response, so that students with little manual dexterity can easily participate. One of the advantages of the program is that the use of the computer takes handwriting out of the equation, allowing students with writing problems (including simply hating to write) to work on reading without having any complicating emotional factors.

Second, the student can be in control of the texts selected for inclusion. Many special needs students, particularly in the upper grades, are tired of the low-level texts they use to practice reading in resource rooms. With a powerful and flexible program, they can bring in materials from any source, prepare the questions, and use selections donated and chosen by their peers.

Third, the teacher can control much of the program. All aspects of the game can be adjusted to individual needs upon loading, so a student with very low frustration level can be given a low total of points needed to win. The sound can be on or off, timers visible or invisible, allowable speed adjusted. Most important, a scoreboard is maintained for the students, so that the teacher can get information on performance without having to be with the student when information is displayed.

Fourth, students can choose learning activities which have meaning for them. They can adjust the program, write questions, select their own readings. Using the printing capacity of the program, they can print out booklets of readings and questions, and they can branch from there into more sophisticated word processing. In short, students using such a flexible and powerful computer tool can become central in their own learning and the learning of others in their classes, rather than being relatively passive recipients of drill and practice activities.

A final word

One resource room teacher grabbed **GAPPER** on the first chance she got. She took it into her school and began to use it with a pair of very reluctant fifth grade readers. Their story is powerful.

The students initially liked using the computer. They enjoyed playing both competitively and cooperatively, and improved their reading speeds and comprehension. There was steady work going on, but nothing dramatic.

ne day they began the laborious process of putting text into the game disk. They developed questions for each other, providing feedback for their right and wrong answers. This took a great deal of time and discussion, and they decided that they had to test their ideas with others. Gradually they began to bring their classmates into the resource room to try out their quizzes and their program. Students from their regular classroom were universally impressed that they were doing such powerful work, generating questions and making up and entering selections. At the same time, the resource room teacher and the classroom teacher recognized that this was the golden opportunity for some serious mainstreaming work. The "computer experts" were invited to bring their completed disk into the classroom for others to use during reading period. There was a predictable rush to the computer during reading, and tremendous excitement as the special needs children taught the others how to use **GAPPER**, watched them read texts selected and prepared in the resource room, and, especially, answer their questions about the stories. On returning to the resource room teacher, one remarked, "I like mingling with the other kids during reading time. Me and the computer taught them a lot today!"

Surely this is one of the goals of reading instruction: a student feeling competent to learn, to work, to share the results of that work. For some of our students, that power will come best through the computer, with the support of appropriate, empowering software.

Rebecca Brown Corwin is Associate Professor in the education division at Lesley Graduate School in Cambridge, Massachusetts, where she coordinates the computers and special needs program. A former fifth grade teacher, she now teaches courses in math education, issues in computing, computers and special education, and curriculum development.

Joanne Bronga, who studied computers in education at Lesley College, uses computers in teaching secondary students in special education in Rhode Island. Her interests include software design and program planning.

Charlotte Reid has been a high school resource room teacher in western Montana for six years. She is currently a part-time faculty member at Lesley College and a candidate for the Certificate of Advanced Study in Interactive Technology at Harvard University.

For further information on computer use in reading programs, contact:

Rebecca Corwin, Joanne Bronga, and Charlotte Reid
9 West Street
Somerville, MA 02144

REFERENCES:

Holdaway, D. 1979. **The Foundations of Literacy.** Gosford, N.S.W.: Ashton Scholastic. (Distributed through Heineman, New York.)

Smith, F. 1976. **Reading Without Nonsense.** New York: Teachers College Press.

ARTICLES:

Mokros, J.R. and Russell, S.J. "Learner-Centered Software: A survey of microcomputer use with special needs students." **Journal of Learning Disabilities,** forthcoming.

Ohanian, S. 1984. "Some Reading Software Zaps Students' Desire to Read." **Classroom Computer Learning.** 5(4):30.

COMPUTER SOFTWARE:

CROSSWORD MAGIC. Mindscape.

GAPPER. Pleasantville, New York: Humar. Relations M. dia.

M-SS-NG L-NKS. Sunburst.

MYSTERY SENTENCES. Scholastic.

MUSICAL COMPUTING

by Fred Schouten

Editors' note: Both of us had heard about various emerging pieces of music software out of the corners of our (tin) ears, but neither had seen a broad review of the possibilities for computer-based music instruction until Fred Schouten's manuscript hit our desks. Schouten combines a sweeping if not exhaustive review of programs and approaches with accounts of his own experience with them in the classroom and in teacher training contexts. Of programs that permit the user to compose and revise their own pieces, Schouten writes, "These programs are revolutionizing music composition much as word processors are changing verbal composition, allowing people to work with ideas rather than being so preoccupied with the mechanics of notation." That was music to our ears.

Many music teachers feel that computers have no place in their classrooms; they believe that the expressiveness of the art is incompatible with the scientific coldness of the computer. With the growing number of inexpensive microcomputers in homes and schools, however, we music teachers should develop at least a modicum of familiarity with the capabilities and limitations of this remarkable piece of technology. Computers will not replace the teacher in the music classroom but, with their capacity to produce musical sounds and quality notation, they can and should be used as an effective adjunct to "live" instruction.

I would like to digress for a moment from the topic at hand and briefly present the perspective from which this article is written. I have been teaching music since 1970, during which time I have taught in all areas (instrumental, vocal, and general) and at all levels (elementary, secondary, college, and adult). Recently I filled a two-year temporary appointment as Assistant Professor of Music Education with the Crane School of Music, State University College of Arts and Science in Potsdam, New York. I am now back in the public school teaching instrumental and vocal music and music theory at the high school level. I have tried to bring this varied background into the ideas presented in the following discourse.

Pros and Cons

What advantages are there to using computers in the music classroom? As a teacher's aide, the computer has several good qualities. It is an individualized instructor that provides a student with the opportunity to work at his or her own time and rate while receiving personalized attention. Students work at a comfortable pace, progressing when a specified level of mastery has been attained.

COMPUTERS IN THE CLASSROOM

Psychologists, especially behaviorists like B.F. Skinner, tell us that the most effective moment to learn is the exact moment when reinforcement is applied to correct responses. At that moment a change in behavior occurs or at least is begun. Computer-assisted instruction (CAI) encourages this immediate learning.

Computers contribute to a high degree of motivation. Today's students are enthusiastic about this technology. Research tells us that CAI is neither more nor less effective than other methods of instruction, but it makes learning fun, and the resulting positive attitudes help to ensure learning. Computers are consistent; they do not get tired and lose their patience. Good software (programs) present logical, sequential instruction, something that is not always evident (or possible) in our classrooms or rehearsal rooms. This combination of consistent, sequential, motivating, individualized instruction is particularly effective for the development of skills in basic musicianship, such as aural training of intervals and harmonic patterns.

There are also disadvantages to computer use in the music classroom. These include concerns such as the cost for procuring microcomputer systems and quality software, the lack of programs available for music learning, and the degree of unfamiliarity with CAI exhibited by most music educators. Although more programs are being developed every day, this process is time-consuming and costly. First attempts at programming CAI for music were limited because the software was designed either by computer programming experts with little or no formal music training, or by music educators with little or no computer programming background. Today, as more teachers are becoming computer literate and better sound-producing hardware (equipment) is being invented, this situation is being ameliorated.

Types of Music Programs Available

There are two categories of uses for computers in the music classroom; they can be used as teaching media, or they can function as tools for composition or sound exploration. Before computer instruction can be considered as an alternative to traditional forms of teaching, course goals and objectives must be identified. Programs may then be evaluated for their "fit" with these goals and objectives. It does little good to have a student participate in a lesson that has no relevance to his or her education. Working with preservice and inservice music teachers, I find that a major concern is a felt inadequacy for evaluating computer materials. The considerations for adoption of CAI into the curriculum are much the same as those for identifying educational field trips, for purchasing new music series books, or for choosing music for band rehearsal; the materials or experience must satisfy an educational need.

Several teaching strategies have been employed by software designers, including tutorials, drills and practice, simulations, and games. By far the majority of available music programs are of the drill and practice variety, with a few additional tutorial packages recently appearing in the marketplace. With systematic evaluation and selection techniques in hand, the music teacher can locate good quality programs to assist in the instruction of general, vocal, or instrumental music.

I frequently use the computer to assist me in teaching high school music theory. I first present material to my students, for example the perfect and major intervals. Next I send the students to the computer for practice in visual and aural recognition of the intervals. I have been using MECC's **Music Theory** program and an Apple IIe computer, because it allows the student to select the intervals to be included in the practice session (for example, only perfect fourths and fifths and major thirds and sixths). The program also allows the student to select the number of problems to be presented during the session. At the conclusion of the set of exercises, the student receives a report of the number tried, the number and percentage of correct responses, and the number of intervals that the student heard more than once before responding. I have found that students who may be timid or slow in class drill often progress much more rapidly when interacting with this drill and practice program. I have observed that this is also true at the elementary level, as well as the college level. I have seen grade school students learning to recognize major and minor scales, identifying intervals and chord quality, and practice simple melodic dictation on the computer. While with the Crane School of Music I taught music education and computer classes. There I witnessed future music teachers learning instrumental transpositions, practicing instrumental fingerings, drilling on obscure musical terms, and sharpening aural music theory skills with CAI. Again, most of the software for these learning tasks were of the drill and practice category.

Several programs also have been constructed to allow sound exploration of musical composition on the computer. Some of these use the individual microcomputer's built-in sound production capabilities; others require additional hardware, such as music synthesizer boards, piano-style keyboards, etc. Depending on the design of the program, students may shape sounds (varying timbre, attack rate, or resonance), explore melodic shapes, design harmonic structures, build forms, and so forth. Many software packages for composition (e.g., **Music Construction Set** or **Master Composer**) allow students to work only within our traditional equal-tempered notational system, while a few, such as **MusicLand** by Dr. Martin Lamb, provide a freer exploration of timbre, melody, harmony, and form. Perhaps the most important advantage of the use of computers for this type of experimentation is that the students are working within a nonthreatening environment; trial and error become a natural part of the learning process. Once a successful product has been composed, the student is able to request a recorded copy (on disk ... e) or a hard copy (printed) of the new work at the press of a key. It is also a very simple process to edit the composition, changing notes or entire sections, varying the timbre, adding or deleting portions, transposing parts, then rehearing the composition. These programs are revolutionizing music composition much as word processors are changing verbal composition, allowing people to work with ideas rather than being so preoccupied with the mechanics of notation.

Integrating Computers into the Music Classroom

After the advantages and disadvantages of computer use in the music classroom have been weighed, course goals and objectives have been determined, and software packages that best fit with our curriculum have been identified, the next step is to integrate this new media into regular classroom instruction. This may be accomplished in several ways. The method of implementation in any classroom depends largely on the number of microcomputers available for use and the number and type of software packages owned by the music department or the school.

One mode for usage is an aide to a teacher demonstration or lecture. Here the teacher uses the computer in much the same manner as a chalkboard, piano, or film projector might be used. If the topic were note reading, the teacher could first explain our notational system, then "run" a typical note-naming drill and practice program (for example, Micro Music's **Pick the Pitch** program, Silver Burdett's **Beginning Music Series**, or **Music: Terms and Notation** from MECC). This would be similar to using flash cards or a chalkboard drill, except that it would incorporate the advantages of color graphics and computer sound generation. We may also wish to teach the process of musical composition. In this case, the teacher could use one of the composition programs as a means for writing and playing the work in progress, talking though the decisions being made in the process. Advantages of this use would be the clarity of the computer graphics and sound production, as well as the enhanced ability to revise, store, play, and print copies of the composition. Simultaneous to the composition technique being discussed, the students also are learning the operation of the composition program, thereby giving them increased probabilities of success when they attempt to compose a piece of music.

Expanding on this idea, it is possible to have a small group of two or three students come to the computer. This group can then participate in a problem-solving activity, such as creating a ternary form using the composition program, while the rest of the class observes, possibly offering suggestions or alternatives to the solutions posed by the small group. This has the added motivation of seeing peers working through the process. Thus, it may help to hold student interest.

There are times when we want to truly individualize the learning in our classrooms. Reasons for this include the need for remedial work, the appetite for extended exploration of the subject matter by gifted or advanced students, or merely the desire to provide alternative experiences for our students. One means to accomplish this is to have small groups of two or three students working with a CAI lesson or composition program in the rear of the classroom as other students pursue optional media. In many schools, the microcomputers are grouped together in a centralized computer lab. This lab can be utilized by individual students or by small groups. Desired computer interaction for music learning can be achieved by reserving the lab for the regular music class time and holding class in the lab. Individual/small group projects can be assigned to be completed in the lab outside of class time. A final, less formal, use of the lab is just to make the class aware of the music programs that are available in the lab. Curiosity will then carry the students to the lab during their spare time; many of these packages are so motivating as to be almost addictive (witness the popularity of computerized game arcades).

Other Uses of the Computer in the Music Room

In addition to teaching and composing applications, there are other benefits to be derived from having the computer in the music room. In the rare instances when there are no students around, the computer can become an invaluable administrative assistant. One of the most popular uses of microcomputers is word processing. We can use word processing programs (e.g., **Speedscript 3.0**) to help prepare press releases, concert programs, class handouts, and those many form letters we use to request product catalogs, to send home student progress reports, and especially to inform parents of plans during recruitment. Once the perfect recruitment letter has been drafted and stored on a floppy disk, it becomes a simple task to change dates on the original letter and press a key to have the new personalized letters printed. Many of the word processing programs allow the simultaneous use of databases. Here the names of the new students, their parents' names, addresses, etc. are entered into the database. When the recruitment letters are printed, the unique student and parent names are placed into the letters where appropriate. I have found that this substantially increases the number of recruits for the band program.

The computer also can be used for record-keeping. Music teachers typically have walls of filing cabinets and index card boxes to store and catalog the music library, uniforms and robes, school instruments, etc. This information is much more accessible and less cumbersome to maintain if done on the computer. Several excellent record-keeping programs, designed especially for the music teacher, are available (for example, the Wenger Corporation's **School Music Manager Series**). Gradebook programs are available to store student data. Many of these programs also allow the teacher to calculate final grades based on the criteria specified by the teacher. Programs such as these free up many hours of the teacher's prep time, which then becomes available for use in lesson planning, or for actual teaching. Calculating final grades for a class of twenty-five students, using weighted grades for various homework assignments, quizzes, and mid-term and final exams used to take me at least an hour. Using a gradebook program reduces this task to a matter of minutes, and I have a printout of individual student grades for the term, a grade summary alphabetically by student, and a simple statistical analysis for the class, including average, range of grades, etc.

Another application of the computer, much like the word processing discussed above, is that of composing, arranging, transposing, and printing music. The same software used in class for assignments (for example, the **Polywriter** system) becomes a potent tool in the teacher's hands. The advantages of these programs discussed earlier apply here, also. Using the computer as a compositional tool and calligrapher again frees additional time for the teacher. As an example of the use of the computer in a performance class, for this year's fall concert I used a Commodore 64 computer and the **Master Composer** program to create the accompaniment for one of the works performed by my high school Concert Choir. The accompaniment, originally written for bass and trumpets, sounds even more exciting when realized on the computer. This composition program allows for full timbral and dynamic control, so the expressiveness of the music is not forsaken.

A final administrative application that is becoming increasingly popular is the use of the computer to assist in designing and charting football halftime marching routines. Several companies are marketing software that plots the band on the field, allows the designer to view the formation from various angles, and prints the charts for dissemination to the ensemble. This is also quite a timesaver for the harried band director. A representative example of this application is Wenger Corporation's **Halftime** software.

Conclusions

It is impossible in an article of this length to cover all of the reasons for incorporating the use of microcomputers into the music classroom. It has been my intent to provide an overview of the uses of this technology especially for instructional purposes. Many music teachers are unsure of the meaning of the term, computer literate. In my opinion, music teachers need to know and have basic skills in three areas: 1) how to evaluate hardware and software in terms of existing course goals and objectives, 2) how to integrate computer learning materials and compositional tools into regular classroom teaching, and 3) how to design instruction for computer delivery or, at least, how to suggest improvements in existing software to publishing companies. These three areas apply to all educational media; teachers regularly examine and evaluate learning materials, integrate the newly-acquired materials into their curriculum, and suggest changes to publishers and authors. The only new factor present in the use of the computers in the music classroom is the machine itself. I do not believe that it is necessary for music teachers to know how the internal functions of the computer operate, nor is it imperative to learn computer programming in order to incorporate the advantages of computer-assisted instruction into music teaching. Music classroom teachers can develop the three computer literacy skills listed above through summer or extension courses offered by local universities or junior colleges, by attending inservice workshops on these topics, or by reading authoritative articles and books. Two books that treat such topics specifically for music educators are **Music and the Apple II** by Thomas E. Rudolph, published in 1984 by Unsinn Publications, Inc., P.O. Box 672, Drexel Hill, PA 19026; and **A Planning Guide to Successful Computer Instruction** by John Eddins and G. David Peters, published in 1981 by Electronic Courseware Systems, Inc., 309 Windsor Road, Champaign, IL 61820.

Once the computer becomes a trusted friend in our classrooms, we will find that we are spending much more time actually sharing with students the excitement of the art form and less time keeping records, drilling minutia, etc. A computer in my music classroom? Definitely!

James F. Schouten, Director of Music at Mazon-Verona-Kinsman High School in Mazon, Illinois, has used microcomputers in the classroom for several years. He has programmed software for use in his music classes. He has also developed software commercially including courseware for the PLATO system. Mr. Schouten has published several articles about learning music. For further information about the use of computers in the music classroom, contact:

James F. Schouten
Assistant Professor of Music
723 N. Charles Street
Shelbyville, IL 62565

Selected Computer Programs for the Music Classroom

Following is an annotated listing of the software (and related hardware) mentioned as examples in the preceding article. It is not meant to be an exhaustive bibliography of exemplary programs for the music classroom, but rather a starting point for music classroom teachers.

Beginning Music I: Reading and Playing Melody. A keyboard overlay transforms the typewriter keyboard into a piano-style keyboard for entry of answers. For Apple II family and Atari computers. (Silver Burdett Company, 250 James Street CN018, Morristown, NJ 07960.)

Halftime software. Computerized charting routines for halftime marching band shows. Very flexible, routines are easily modified and edited, allows perspective viewing. Any screen may be printed (hardcopy). For Apple II family and TRS-80 (Radio Shack) computers. (Wenger Corporation, P.O. Box 448, Owatonna, MN 55060.)

Master Composer. A music composition tool software program. Compose, notate, (modified traditional notation), edit, and set parameters for three voices. Save compositions/voicings on disk and use with other programs; print compositions. Help screens available at any time. For the Commodore 64 computer. (Access Software, Inc., 925 East 900 South, Salt Lake City, UT 84105.)

Music Construction Set. A music composition tool software program. Compose, notate (conventional notation), edit, and play one to three voices. Can be used with Koala Pad or joystick. Suitable for young elementary students through older, advanced music students. Compositions may be printed. For Apple II family computers or the Commodore 64 computer. (Electronic Arts, 390 Swift Avenue, South San Francisco, CA 94080).

Musicland. Four screens allow creativity and exploration: Music Doodles to create musical shapes and small thematic blocks, Timbre Painting to "orchestrate" the Doodles, Music Blocks to design form with the Doodle blocks, and Sound Factory to build timbres. Uses joystick or Koala Pad for entry—no typing required. Plays through the Mountain Music System stereo sound synthesizer. For Apple II family computers. (Syntauri Corporation, 1670 South Amphlett Boulevard, Suite 116, San Mateo, CA 94402.)

Music: Terms and Notation. Teaches and drills note names in the treble and bass clefs (with ledger lines and accidentals), note values, and musical terms. For Apple II family and Commodore 64 computers. (Minnesota Educational Computing Consortium (MECC), 3490 Lexington Avenue North, St. Paul, MN 55122.)

Pick the Pitch. Drill and practice for recognition of notated pitches. Has tutor and game modes for skill development. Plays through the Micro Music DAC (digital-to-analog conversion) synthesizer board. For Apple II family computers. (Temporal Acuity Products, Inc., Building I, Suite 200, 300-120th Avenue NE, Belleview, WA 98005.)

Polywriter software. Polyphonic chord notation, traditional notation on printout, automatic instrumental transpositions, editing and rewriting capabilities, enter music from the five-octave keyboard. For use with SOUNDCHASER SYSTEM, Apple II family computers. (Wenger Corporation, P.O. Box 448, Owatonna, MN 55060.)

School Manager Series. Available programs include: MUSIC BUDGET MANAGER, COMPUTERIZED GRADEBOOK, UNIFORM INVENTORY, INSTRUMENT INVENTORY, MUSIC LIBRARY MANAGER. Versions available for Apple II family and IBM PC computers. (Wenger Corporation, P.O. Box 448, Owatonna, MN 55060.)

Soundchaser System. Five-octave keyboard, synthesizer board, software for recording musical compositions. For the Apple II computer family. (Wenger Corporation, P.O. Box 448, Owatonna, MN 55060.)

Speedscript 3.0 software. Inexpensive word processor that contains many features available on higher-priced packages. Create, edit, and print documents; save documents on cassette tape or floppy disk for future use or revision. For Apple II family, Atari, and Commodore 64 and Vic 20 computers. (COMPUTE! Publications, Inc., P.O. Box 5058, Greensboro, NC 27403.)

IV LEARNING TO PROGRAM AND PROGRAMMING TO LEARN

THREE YEARS WITH LOGO IN A 3RD GRADE CLASSROOM

by Linda Benedict Colvin

Editors' note: The special value of Linda Colvin's article is captured in her subtitle. It is the story of her experience in and out of the classroom over three years, with starts and stops, missteps and second thoughts, increasing confidence in her own judgement, and mastery gained through trial and reflection. While Colvin learned a great deal about the sequencing of Logo activities in a third grade classroom, and we can certainly profit from the specific lessons she learned, perhaps the more important lesson can be learned from her example as a reflective, persistent practitioner. She appropriates the ideas and recommendations of the Logo experts but makes the insights her own by refashioning them in the light of her own classroom experience.

In the summer of 1982, when I participated in a Research and Development Project for the Walpole Public Schools, aimed at developing a K-4 computer literacy curriculum, the Logo language had just become available for the Apple computer. There was little information available, at that time, to help guide our group in making decisions about what Logo activities to teach at the elementary level. As the person in charge of researching and writing up a suggested Logo sequence, I spent much time contacting people and organizations across the country who seemed to be leaders in developing computer literacy curricula. It became evident to me, that (1) practically nothing was available in writing—especially regarding how to teach Logo, (2) that the best source of practical ideas was "teacher-students" who had taken Logo courses, and (3) that people "out there" were eager to hear from me about the results of my own project.

Ultimately, I developed a sequence based primarily on Dan Watt's Brookline Logo Curriculum,[1] extensive discussions with Louisa Birch based on her experiences at the Meadowbrook School in Weston, and Logo Course projects by Carol Shore of Providence, R.I. and Dot Pelc, a Lesley College graduate student. I did not attempt to specify precisely which activities needed to be engaged in at which levels. Instead, I wrote a continuous sequence which I felt would lead children gently and very gradually into dealing with the Logo language.

The original sequence was based on an assumption that, since the idea of computer languages was so new and seemingly difficult for adults to grasp, it would be equally difficult for children to deal with. Based on suggestions in Louisa Birch's pilot curriculum for K-3[2], children would be gradually exposed to the turtle and its capabilities through **Instant.**

Instant, a simplified version of Logo written for younger children, allows them to explore the world of the turtle using individual keystrokes. In **Instant** 1 F (rather than FD 10) moves the turtle forward 10 steps, R is right (30 degrees), L is left, S produces a square, C a curve and E erases. **Instant** #2 adds in B-ack, T-riangle, H-rectangle, P-en up and down, and U-ndo. **Instant** #3 adds colors for the pen and **Instant** #4 provides an opportunity to name a picture, save it and ask for it later on.

The use of **Instant** was to be followed by **Delta Drawing** where children would, again, gradually be introduced to various language concepts—this time with more emphasis on use of procedures within superprocedures, the various screens (text, graphics), and the ability to edit programs. The Walpole computer literacy curriculum, which grew out of our summer project, did not specify that children would be learning "pure, unadulterated Logo" until they reached the third grade level.

The actual logo sequence, when it was reached, was to include the following:

1. FD, BK, LT, RT, DRAW.
2. PU, PD, HT, ST, BG, PC.
3. REPEAT.
4. Procedures.
5. Superprocedures.

Almost as an afterthought, I added in:

6. Filing and managing workspace.
7. More detailed edit commands.
8. Variables.
9. Recursion.
10. Conditional expressions.
11. Naming, e.g., make

At the time this was originally developed, I had not taken a Logo course myself and was somewhat unclear as to what #6 to 11 above actually meant. (Hoping, certainly, never to have to teach these to someone else!) I also remember being quite surprised when, in commenting on my final project for a computer literacy course, the instructor suggested that some children at the elementary level might even be introduced to game structures such as those in **Shoot** and **Instant.**

In retrospect, it is obvious to me that educators' assumptions about children's ability to learn Logo, and to deal with computers in general, were severly limited by our own "computer phobias." Because we, as teachers, were apprehensive about this new field, we assumed that children would be as well. Little did we realize the impact the computer's presence would have on the development of children's thinking and interaction.

Walpole's official K-12 Computer Curriculum Committee, for which I was an advisor during the 1982-1983 school year, incorporated most of our suggestions into the suggested sequence, specifiying that at grade 4, the highest level, students should begin to use REPEAT when applicable, should understand how to write and debug a procedure in Logo, and should be able to use variables and recursion in procedures. When this was first distributed to teachers, the reaction was one of panic especially vis-a-vis the Logo sequence. Although third and fourth grade teachers had gained some limited exposure to the Logo language through a course I had given, they did not feel sure enough to teach it themselves. Many found it difficult to accept our suggestions that they could learn alongside the children, and until given a short, intensive Logo course this year, left much of the Logo teaching to me.

The 1982-1983 School Year: Learning Through Mistakes

Realizing that if I was to have any legitimacy as a proponent of the Logo language for elementary school students, I would need to begin to teach it myself, I signed up for my own first full-semester, intensive Logo course in the Fall of 1982. Prior to this, I had only had a "taste" of Logo as a short unit in a more general computer literacy course. My primary goal in taking this course was to begin an informal pilot project in my third grade classroom to teach Logo. As I was apprehensive about my own ability to understand the Logo language, I started out very cautiously, keeping the development of my program under a number of strict controls.

The following aspects of my first tentative pilot in teaching Logo illustrate my own limited vision of what children could understand.

I had the computer in my classroom only one day a week for the first half of the year. This was partly dictated, at the very beginning, by our having only one computer in the building. Later, however, when more computers were available, I still kept fairly strict control over the learning by continuing to specify only one day per week of ten to fifteen minute periods for pairs of children.

I spent a great deal of time, in retrospect, on off-computer activities, as these were something I had done before and felt comfortable with. Examples of these include the use of a "Computer Literacy" booklet of five lessons I'd developed in 1980, special parent interviews on computer applications, reading of **Katie and the Computer**[4] and development of flow charts.

Although these can have an important place in a computer literacy curriculum, I feel that it is important for teachers not to hide behind these more traditional book-learning activities. With only a limited amount of time officially available under our time allotment for computer literacy activities, we must be sure we are not defeating our overall purpose by not ever actually letting the children work at the computer itself. Too much padding of the curriculum with noncomputer activities may end up aborting what we have set out to do.

It was not until late October that I began to introduce children to what I saw as a modified, child-appropriate version of the Logo language. When I introduced **Delta Drawing** to my class for the first time, I made a note in my journal that "I had failed to anticipate that the children's drawings—with only a few commands—would be so sophisticated that I would want to save them . . . To me, the projects were amazing in the amount of planning and control over the computer that they reflected." Although I was ecstatic about the children's response, I became even more apprehensive than before over my own ability to control and keep ahead of the children's learning.

The following week I introduced the Graphics vs. Text screen, as well as some information about how to save pictures on disks. In November, I introduced the REPEAT keys as well as concepts of using procedures (or programs) within other programs. After spending a month with **Delta**, however, I began to realize its limitations as a lead-in to Logo itself. These included 1) the difficulty children were having in seeing the structure of their programs when looking at the Text screen, 2) difficulty in editing programs or changing them, 3) difficulty in achieving fine control over what the turtle was doing, and 4) my own difficulty, as a teacher, in getting an idea of how the child's mind was working when drawing a picture, as the savepict and readpict commands did not redraw the picture as the children had developed it.

Finally, in December (at the end of my own Logo 1 course), I began to introduce "real Logo"[5] to my class. Initial reaction, after the incredible ease of **Delta**, was one of frustration. There seemed to be so much more to remember and the children had, in a sense, been spoiled by their **Delta** experience, rather than helped. I felt discouraged and ready to give up before I'd even begun.

Knowing from personal experience how exciting and powerful the Logo language could be, I determined in my New Year's resolutions for 1983 to make Logo work for me and my children. To do this, I started to have the computer available in my classroom for several weeks at a time. I gave the children a variable circle, square and triangle procedure (not expecting them to be able to understand how it was developed themselves) to use as a tool in creating programs. I also suggested the POLY procedure to them as one way to begin to explore some of the neat phenomena of the Logo language. POLY enabled them to begin to learn, through experimentation, about the effect changing the angle can have on a closed polygon.

As a result of my efforts during the spring of 1983, the children finally began to feel comfortable with Logo itself. That first year, I made many mistakes. My revisions often are tailored to the needs of my class. Mistakes I made my first year with Logo included:

1. Giving every pair of children their own disk to start with. Although this was exciting for them, they experienced a great deal of confusion over the naming of files on disk vs. procedures in working memory. They often either accidentally wiped out things on disk, by not reading in their files to start out, or else had so many files on disk that they would spend half of their work time trying to find what they had done before. This is part of a phenomenon which I will call trying to give them everything at once.
2. Another mistake was not allowing children enough time to simply explore the various things the turtle could do—and record what they had discovered—before plunging them into teaching the turtle.

3. I tried to introduce them to all of the editing commands at the start (all of the <ctrl> key possibilities) and as a result, the children were very confused over what to use and how.
4. A final problem was one of trying to introduce some concept as a whole class lesson—e.g., teaching the turtle—when only about ⅓ of the children were ready to grasp it.

Changes from 1983-present: Exploring Unforeseen Horizons

My first experiences teaching Logo in 1982-1983 led me to make some drastic changes in teaching strategy and attitude which I continue to follow to the present day, with minor modifications. As a result of these changes, I am continually amazed by how much children can absorb, and how quickly, when I provide them with the right ideas at the right time. This has required much more involvement, on my part, with what the children are doing every day at the computer. Some of this has been enhanced by a computer lab opening at the Fisher School, afternoons only, in December of 1983.

The following are some of the things I have done to improve the Logo learning experience for my class:

1. **I have allowed children to choose their own partners**, from the very start, and encouraged them to try to develop a cooperative and continuing working relationship with their chosen partner. This has cut down on the amount of time spent arguing with an incompatible Logo partner over whose turn it is, what to focus on for that period, etc.
2. **I now begin Logo at the very beginning of the year**, cutting out all of the six months of "preliminary introductory" activities—including **Delta Drawing**, computer literacy in general, **Big Trak**[6] since all of my children are coming to third grade having had experiences at the second grade level with **Instant**. This reflects the overall effects of having begun to implement a complete K-4 computer literacy curriculum during the past year.
3. **I have allowed much more time for exploration** of, and recording of, the turtle's primitives—including 2 to 3 weeks "messing around with" the REPEAT command. I have also allowed for a lot of exploration of pen and background color, which I was unable to do much with the first year as we had only a black and white monitor with our old Apple II+. I feel it is important for the children to have discovered a number of things which the turtle can do, before they begin to worry about all that is involved in actually "teaching the turtle."
4. **In introducing the children to teaching the turtle procedures,** I waited until I felt the majority of the children were able to grasp what I was talking about. (I did pre-teach the new mode to assistant students who had computers at home so they could help with further class instruction.) Then, I introduced the class to new rooms in the turtle's house; I compared the new

edit mode to the turtle's study or workroom and the old immediate mode to his playroom.

I kept the editing commands to the use of the right and left arrow keys and <esc>. I sat down with **every** pair of children the first time they taught the turtle something they had worked out previously, rather than letting them stumble around trying to remember and use what I had taught them in the whole class lesson.

5. **In terms of use of disks,** I started out with a **class disk** on which children would save their work under their own names (instead of a disk per pair). Instead of trying to introduce all of the class to disk handling (file naming, reading, saving, catalog, etc.) at the same time as teaching them about procedures, I took aside one child who had a firm grasp of what I was talking about and trained him or her to do this for the other during the first week. This worked well, although many wanted to be able to do it all by themselves the following week.

During the past two years, I have done a lot more with incidental teaching, telling children about how to use different commands as the need arises. This has been used in teaching such concepts as the various screens (text, splitscreen, fullscreen), more sophisticated editing commands, the use of variables and the use of recursion.

I find that, as a result of teaching hints to various groups as the need arises (rather than doing things as a whole group): 1) the children understand and try to use what I tell them right away, because it is directly relevant to their immediate needs, 2) the underground culture that exists among the children spreads the teaching of these specific tricks to others because it is something neat that they know and the others don't, yet, and 3) eventually, there is enough of a demand—need to know—among the class as a whole that I will make the teaching of a particular concept—e.g., variables—official, through a whole class lesson and the posting of a hint card for use by the class.

Developing the Logo learning sequence from the individual children's needs, rather than from my own perception of what they ought to know at a particular point in time has proven a **much** more powerful and meaningful teaching strategy for me.

Because I have allowed for many more individual paths of growth, as well as having provided the children with a computer in the classroom every day, I have begun to realize that there are few limits on what children of varying abilities can grasp, other than those I percieve in my own understanding. Long ago, I came to realize that with Logo, children who are ordinarily seen as handicapped in their academic abilities find paths of understanding within the language I had never anticipated. There are, of course, some children who need to spend more time with simple spaghetti style exploration, some who don't feel as much of a need to extract patterns (e.g., of repetition) from what they are doing in order to make the turtle more efficient, and some who need to have concrete ideas presented to them for possible projects. It is not as easy, though, to associate certain types of planning and growth with certain ability levels, as it may be in other subjects. The partners the children are working with, their own frame of mind on a particular day, and their own ideas about what they want to accomplish and what is relevant to them may affect their level of development at different points in time.

Finally, as a result of my experiences working with children learning Logo since 1982, I have found the following specific Logo concepts (which I had not felt children could grasp at first) to be relevant and understandable.

1. **The use of variables.** After several weeks of exploration with learning to teach the turtle various shapes and a growing desire to be able to more easily modify the sizes of various shapes in order to use them, I found it a natural to introduce children to the idea of variables. I started by brainstorming a list of real life variables (height, weight, ability, and interest). Then, while working with the students to develop procedures for a little square, medium square, and big square, we looked carefully at what specific command was being changed each time to change the size. Although the :side (or :size) terminology was somewhat abstract (a black box), they were able to under-

stand how to use this and incorporate it into their procedures within one or two sessions of exploration.

2. **The use of recursion.** This developed with one group at first as they were working on a blizzard superprocedure. The children had first taught the turtle how to make snowflakes. Then they decided to type in exactly the same set of directions over and over again to create their blizzard. I suggested to them that they could just type in blizzard at the end, thus getting it to call itself repeatedly. Their response was. . .of course!
3. **The use of a stop procedure in recursion.** This became necessary when I introduced a POLYSPI microworld and children began to want to use it in other procedures. The POLYSPI procedure we used allowed for variable inputs to color, side and turn. Because the side was increased by two turtle steps each time the procedure called itself, the children were able to fill the entire screen with spirals of various shapes. They devised some fascinating ways to record and fiddle with the variables in the procedure (including inputting birthdates). I explained to them that they could simply add in a line that said "If :s > 100 then STOP." Again, although the :S(ide) was somewhat abstract, they could understand the need for it and how it worked.
4. **The use of print.** Some children, who had had some exposure to Basic at home, wanted to do more than just turtle graphics when their turn came at the computer. Because I happened to like graphics, I had not introduced the class to the print capabilities in Logo. However, as I noticed one group typing in their names and various letters of the alphabet, I decided I could show them how to write a procedure to print out their names on the screen. As soon as I had done this, one of my children asked if it was possible to put this in a for-next loop (would you believe!) so the names would repeat several times. I then showed them how they could write a variable procedure which would allow for the number of times to print out their names as an input. A counting line (:times = :times +1) and a stop procedure in the top line (if :times = 10 stop) also had to be included.
5. **The use of the printer,** on a regular basis, to print out pictures of procedures, as well as the procedures themselves. The breakdown of our school's portable large screen monitor made it necessary to find another way for children to share what they had been doing. As a result the printer was brought into the room on Thursdays and the children began to watch what I was doing in order to understand how to do it themselves.
6. The **introduction of game structures.** Interest in beginning to create games and activities for other students has been unexpected and exciting. It began last year when a gifted student, Danny, got a computer at home in December. He spent a lot of time working with his father with Logo activities. Within a month after he had acquired his own Apple, he was eager and interested in going beyond just teaching the turtle how to draw pictures. At first, I thought that he was coming to class merely parroting things his father had

told him, but upon closer examination and a lot of extra time spent sitting down with him during recesses, I came to realize that he had a true grasp of what he was doing. The "games" he developed included the following:

- **A conversational procedure about E.T.** which included conditional statements, the use of request and, based on the user's response, falling through into another procedure.
- **A soccer game** where a soccer field was drawn on the screen, the turtle placed at home (in the center) and the user allowed to type in turtle primitives until reaching the edge of the screen where NOWRAP had been used in the procedure to give the message to the user "Turtle out of Bounds." As the soccer game became more sophisticated, I introduced Danny to some of the following ideas: the possible use of SETXY and the turtle grapher to make the drawing of the soccer field easier from the top down; the addition of comments next to various parts of the program, to allow him to find and debug sections more easily; the way in which the drawing of the soccer field could be subdivided into specific procedures to draw the end lines, goals and center circle—all of which would become part of the bigger procedure. Finally, Danny became interested in limiting the "acceptable choices" of commands (although I, personally, had liked the original version where the user could freely explore the boundaries using regular Logo commands). He wanted to print out choices, if the user so desired, and then access an **Instant**-like procedure.

Each year, I find more and more children progressing beyond the graphics exploration aspect of Logo to relating the things they are doing with Logo to their everyday needs. One group of children used Logo to develop a picture of Saturn for the cover of their planet report. Another group created a Tic Tac Toe game for their classmates to play in their spare time. In social studies, after learning of different types of communities, children created procedures to draw villages, suburbs and cities (with varied repetitions of houses). After a two-week unit on graphing, Scott decided to try to transfer his graphed sports car onto the Logo screen using the X-Y coordinate grid. I believe teachers and students need to begin to find more and more ways to build bridges between the world of Logo and the real world curriculum.

Conclusion: Where Do We Go From Here?

My ideas of what children are able to do—and when—with Logo have changed dramatically during the past 3 years. The speed with which children are able to absorb Logo has been affected by a number of different factors including: my own ideas about how to go about teaching the language; the increasing amount of time children have access to computers—both at school and at home; and my own knowledge of Logo—including a better grasp of recursion, game structures, variables, and other sophisticated components of the language. A number of children seem ready and eager to learn as much as I can give them as fast as I am able to get my own thoughts together on how to introduce it. For me, this is remarkably exciting but time-consuming. I find myself lying awake at night spending as much time thinking about how I can help various children deal with their programs as I used to with my own programs.

It is important to realize, however, that for many teachers and parents, the rapid growth of their children's Logo knowledge is frightening and unmanageable. Establishing a set curriculum of what should be taught and in exactly what order seems like a nice way to try to keep everything under control, but as Papert has noted regarding the Lamplighter experience,[7] the underground culture among the students about the powerful ideas of Logo will soon undermine our efforts in this direction. I am beginning to feel that we, as educators, need to reconsider how we are structuring our computer curriculum. We are setting up goals and objectives based on old ideas about teaching and learning. To me, the computer, and especially the Logo language, will prove that old ideas about teaching may no longer work for a future, more technological society.

The solution as I see it, is for us to continue to expand our own knowledge about Logo (as well as other computer tools) and, as rapidly as possible, make this accessible to students who require it. The curriculum will need to become more continuous and individualized. It may also be necessary, at some point, to realize that perhaps not every teacher needs to be an expert in the Logo language. A network of those who are interested, however should be developed to keep on top of new ideas, as the field is spreading faster than the ability of any one of the experts (myself included!) to keep up with it.

Linda B. Colvin is chairman of the department of Computer Education for the Walpole Public Schools in Walpole, Massachusetts. She has been an interested user of microcomputers since 1976 and has used them on a daily basis in her classrooms for five years. In 1984 Linda was awarded a Certificate of Merit in **Electronic Learning's** *annual "Educator of the Year" issue. For further information on her program with third graders, contact:*

Linda B. Colvin
One Knollwood Street
North Easton, MA 02356

REFERENCES

1. Watt, Daniel H. June 1981. **Logo Curriculum Materials Specially Adapted for the Brookline Public School,** M.I.T., Cambridge, MA.
2. Birch, Louisa. 1984. **Introducing Logo to Primary Grade Children,** Houghton-Mifflin Co., Boston, MA.
3. Computer Access Corporation. 1981. **Delta Drawing, Version 1.0,** Medford, MA.
4. D'Ignazio, Fred. 1979. **Katie and the Computer,** Creative Computing Press, Morristown, NY.
5. Terrapin, Inc. 1982. **Terrapin Logo Language,** Cambridge, MA.
6. **Big Trak,** a programmable toy developed by Milton Bradley Co.
7. Watt, Dan. August 1982. "Logo in the Schools," **Byte Magazine,** Vol. 7, No. 8, Peterborough, N.H. Also, a talk delivered by Seymour Papert at M.I.T. in the spring of 1983.

BIBLIOGRAPHY

Ableson, Harold. 1982. **Logo for the Apple II,** Byte/McGraw-Hill, Peterborough, N.H.

Birch, Louisa. 1984. **Introducing Logo to Primary Children,** Houghton-Mifflin Co., Boston, MA.

Coburn, P., Kelman, P., Roberts, N., Watt, D. and Weiner, C. 1982. **Practical Guide to Computers In Education.** Addison-Wesley Publishers, Reading, MA.

Computer Access Corporation. 1981. **Delta Drawing—Version 1.0,** Medford, MA.

D'Ignazio Fred. 1979. **Katie and the Computer,** Creative Computing Press, Morristown, NY.

Feurzeig, W., Horwith, P. and Nickerson, R.S. 1981. **Microcomputers In Education,** Report No. 4798, Bolt, Beranek and Newman, Cambridge, MA.

Johnson, Anderson, Hansen and Klassen. February 1980. "Computer Literacy: What Is It?" in **The Mathematics Teacher.**

Lowd, Beth. "Guide For Parents Volunteering to Help With Logo," paper submitted for a Lesley College Course.

Luehrmann, Arthur, "Computer Literacy: What It Should Be," submitted to **Mathematics Teacher.**

Lynch, Charlotte, "Introducing Computers in the Elementary School," in **Computing Teacher,** Vol. 8, No. 2.

Lynch, Charlotte. 1981. **Pilot for an Elementary Curriculum on Computer Literacy and Programming,** Andover Public Schools MECC Services, **Apple Logo in the Classroom,** MECC, St. Paul, MN.

Northwest Regional Educational Laboratory, **K-12 Course Goals In Computer**

Literacy, Portland, OR.

Oliver, Joan, "Observations of Logo Programming In The Classroom," paper submitted for a Lesley College Course.

Papert, Seymour, Watt, Daniel, diSessa, A., Weir, Sylvia. September 1979. "Final Report of the Brookline Logo Project: Part II," M.I.T. Artificial Intelligence Laboratory, Cambridge, MA.

Papert, Seymour. 1980. **Mindstorms: Children, Computers and Powerful Ideas,** Basic Books, Inc., NY.

Pelc, Dot, "Lesson Plans to Introduce Logo to Elementary School Students," paper submitted for Lesley College Course.

Riordan, Tim. November 1982. "Creating A Logo Environment," **The Computing Teacher,** Vol. 10, No. 3, ICEE, Eugene, Oregon.

Seidel, R., Anderson, R. and Hunter, B., (editors). 1982. **Computer Literacy,** Hunter Academic Press, NY.

Shore, Carol. "Sequential Activities for Beginning Logo Users," paper submitted for Lesley College Course.

Terrapin, Inc. June 1982. **Terrapin Logo Language,** Cambridge, MA.

Walpole Public Schools K-12 Computer Curriculum Committee, "Goals for a K-12 Computer Literacy Curriculum."

Watt, Daniel H. 1983. **Learning With Logo,** Byte/McGraw-Hill, NY.

Watt, Daniel H. August, 1982. "Logo in the Schools." **Byte Magazine,** Vol. 7, No. 8, Peterborough, NH.

Watt, Daniel H. June 1981. **Logo Curriculum Materials Specially Adapted for the Brookline Public Schools,** M.I.T.

Watt, Daniel H. Winter 1981. "Computer Literacy: What Should Schools Be Doing About It?" NESDEC Exchange.

Watt, Molly. "What Is Logo?" **Creative Computing.**

PUTTING THE FUN BACK INTO RECURSION

by Molly Watt

Editors' note: Even people who doubt there was any fun in recursion in the first place will believe Molly Watt has put it back with the dramatic exercises she describes here. Whether you and your students have ever been lost in recursion's hall of mirrors or have never had the pleasure, Watt's "recursion plays" will provide you with a way of thinking about this powerful but elusive concept. If you teach programming, you may not only find Watt's scripts useful for teaching recursion, but suggestive of similar ways to teach simpler programming concepts in Logo as well as in other languages. In fact, if you teach with any of a broad range of tool programs, you may find Watt's dramatic techniques useful. Imagine, for example, the script for a search within a database management program.

Teaching the powerful concept of recursion is a fundamental job for all teachers teaching with Logo. Recursion is a powerful mathematical process which involves "a chain of deferred operations" (Abelson and Sussman, 1985). It is also a particular syntax for writing Logo programs. Because the concept of recursion is abstract, not commonly apparent in everyday life, teachers find it a foreign way to think. Therefore, it becomes a stumbling block in both their own use of Logo and their teaching of their students.

"Playing turtle" by becoming the turtle and walking a shape before going to the computer to draw it provides a tool for thinking about using Logo (Papert, 1980). In the same way, producing plays that become metaphors for how Logo works provides Logo learners of all ages images of how Logo recursion works and an appreciation of its power.

My experience confirms the observation that teachers teach as they are taught. In our work with teachers learning Logo at Simon Fraser University, Keene State College, and Lesley College, Dan Watt, Tony Stavely, and I have included images of recursion from our first sessions. I start with the simplest case of a recursive description of an iterative process (Abelson and Sussman, 1985) to a mutual, indirect recursion embedded in a superprocedure. I use a mix of people procedure plays and actual Logo procedures as scripts (M. Watt, 1985 c). The process of participating is a social and lighthearted classroom adventure. I notice that the plays have made it possible for all of my students to really get the idea clearly and easily, where previously confusion abounded, and many students, usually women, declined to attempt full understanding or use of recursion. I believe the process of using plays is successful because it allows the teacher/learner to think recursively before attempting to write a program using recursion and the correct and specific Logo syntax.

This article is a detailed account of how I actually build the concept. Because it uses words to describe what is essentially a process, the experience which sparkles with conviviality and "aha" and creating together is missing from the article. Attempt to create an image which includes this while you read.

My first thoughts about recursion came long before I encountered Logo. I was intrigued with a picture in a magazine. It showed a girl about my age holding a picture of herself holding a picture of herself holding a picture of herself. . . .The picture never finished. It simply became too small for my eyes to differentiate. The spiralling of this example continues to delight me.

The idea of recursion is simple: there is a big job to do and Logo tackles it in a deceptively easy manner. Take one of my versions of the classic people procedure I first heard about from Seymour Papert. Here is how

```
TO WOW
STAND UP
RAISE YOUR ARMS
LOWER YOUR ARMS
END
```

I solicit a volunteer to act out WOW by following the procedure script one line at a time. When the volunteer gets to the line "END," I intervene with the instruction "END by telling me that you are done." This in effect returns command to top level. The volunteer does this, perhaps a little sheepishly, sits down, we applaud, and the fun has begun.

Now add the numeral 2 to the name of the WOW procedure to edit its title and a new line to illustrate tail recursion. Here is how

```
TO WOW2
STAND UP
RAISE YOUR ARMS
LOWER YOUR ARMS
WOW2
END
```

I ask for a volunteer to follow the revised script. Usually when the person reaches the second WOW2 line they call themselves WOW2 and start to continue through the same script a second time. At this point I intervene. Looping back does not create the most powerful image for thinking about tail recursion. Logo is asking for another copy of the procedure WOW2, and so must the volunteer by pointing to another actor for the WOW2 play. (The first volunteer must remain standing because END, the last instruction has not been reached yet, and they are not done.) When actor #2 reaches the recursive call, WOW2, the actor must point to a new vounteer to WOW2. This process continues until several people-WOWs are standing, waiting to finish, most of those watching are involved and smiling and it's clear that the point has been indelibly made. I employ the interrupt function. END is never reached by any person, but the play is over (M. Watt, 1985 c).

Teachers new to Logo are clearly amazed to see the results of one small, somewhat elegant procedure. Of course this just starts the job of teaching recursion. Logo, being much smarter than human beings, has its own convention for getting a job done. Not only does Logo take on a whole, large job and then do only one small piece of it before calling on a helper to do the rest (D. Watt, 1983 b), but Logo usually checks to see if there is anything to do before starting the job. If there is nothing to do, and the conditionals are met to STOP, Logo stops the whole procedure right there and returns to top level.

The idea of recursion is simple: for example, take the tedious job of doing the dishes from a birthday party. The usual way the dishes get done is by the parent of the birthday person after all the guests have departed! But if Logo could leave the computer and take on the job, it would use a recursive procedure to complete the task in a much less burdensome manner. Study the following example:

```
TO DO.BIRTHDAY.PARTY.DISHES
      If your placemat is empty STOP
      Otherwise: clear dishes from placemat
      Wash dishes
      Dry dishes
      Put dishes away
DO.BIRTHDAY.PARTY.DISHES
END
```

We can consider each Logo procedure a script for a play in Logo (Watt and Watt, 1985). Using the above procedure as a play script, the first player takes the role of DO.BIRTHDAY.PARTY.DISHES. The first line of the script directs the player to check to see if the placemat is empty. If there are no dishes to do, the play is over at the end of the first script line. If, however, there are dishes on the placemat—and in our example, imagine that there are indeed—the player continues to follow the procedure script by following its instructions line by line. First the player clears his/her own dishes, then washes the dishes, then dries the dishes, then puts them away and then calls the next player to follow another copy of the procedure script. The next player acts out the same instructions and calls another player who follows a new copy of the same instructions. In this way, one set of the birthday party dishes are completed by each player in turn, following a procedure which includes calling the next player, until eventually as the play moves around the table a player is called who has an empty placemat and the procedure stops and the play ends.

In the same way, students can act out a procedure play of DO.BIRTHDAY.PARTY.DISHES as an example of Logo's wizardry at work. They can begin to understand recursion by playing turtle in a simple tail recursive Logo procedure.

How do you teach the turtle to make a circle in Logo? Well, the simplest answer in English is to tell the turtle to "move forward a little and turn a little." I make many copies of the instruction, "move forward a little and turn a little," and give one copy to the player being the turtle. When the player has completed it, I give another copy of the instruction and when the player has completed that, I give another copy of the same instruction. Eventually the turtle-player has walked an approximation of a circle and accumulated a large sheaf of copies of the command. END has never been reached, and I interrupt the process (M. Watt, 1985 a).

Translating the same example into a Logo procedure would look like this:

```
TO CIRCLE
    FORWARD 1
    RIGHT 1
CIRCLE
END
```

The image is that the first copy of the Logo procedure CIRCLE moves the turtle FORWARD 1 and then RIGHT 1 and then calls for a new copy of the CIRCLE instructions which move the turtle FORWARD 1 and RIGHT 1 and then calls for a new copy of the CIRCLE instructions until the process is interrupted. Note that the instruction END is never reached in CIRCLE, as a new copy of CIRCLE is called just before END is reached. A more complex script for a procedure play is:

```
TO COUNTDOWN :NUMBER
    IF :NUMBER = 0 STOP
    PRINT :NUMBER
COUNTDOWN :NUMBER-1
END
```

(Abelson, 1982)

In this play the first player, COUNTDOWN, receives a number input of any number. For the sake of a simple written example, I'll use a number 2. (In a workshop of people acting out procedures a larger input number is more fun to use!)

COUNTDOWN carrying 2 checks to see, is 2 = 0? It's not. COUNTDOWN carrying 2 goes on to the next line and follows the instruction to print its number, a 2, on the space provided. (I use a pad of paper to simulate the computer monitor.) Then COUNTDOWN carrying 2 subtracts 1 from its 2 and calls a new player COUNTDOWN and hands over an input of 1.

COUNTDOWN carrying 1 checks to see, is 1 = 0? It's not. COUNTDOWN carrying 1 goes on to the next line and follows the instruction to print its number, 1, on the space provided. Then COUNTDOWN carrying 1 subtracts 1 from its 1 and calls a new player COUNTDOWN and hands over an input of 0.

COUNTDOWN carrying 0 checks to see, is 0 = 0? It is. So COUNTDOWN carrying 0 stops and calls, "I'm done," to COUNTDOWN carrying 1 who calls, "I'm done," to COUNTDOWN carrying 2 who tells the audience, "I'm done."

The play is over and applause is in order. After the applause has died down, I execute the same procedure on the computer.

The idea of tail recursion is now fairly well established as an image. I use an edited version of the original WOW procedure to demonstrate embedded recursion and a stop rule. Here is how

```
TO WOW3
IF 5 WOWS ARE STANDING STOP
STAND UP
RAISE ARMS
LOWER ARMS
WOW3
SIT DOWN
END
```

If 5 WOWS ARE STANDING, STOP, mirrors the logo convention of seeing if you can stop before actually doing any work! It also gives another opportunity to talk about stopping a procedure when a conditional has been met. This WOW3 procedure can also stop if the instruction END is reached.

The second edit in this procedure is the addition of the command SIT DOWN following the recursion line WOW3. I ask teachers to predict what this procedure script is instructing its players to do before we act it out. Some say you'll never get to END, while others disagree, citing the conditional line. Consensus has never been reached!

Once again we act it out, player 1 following the instructions to the point of calling a new player, but still holding a list of SIT DOWN and END not executed, and waiting while player 2 holds another copy of the same list and gets to the WOW3 recursion line and calls player 3, with commands SIT DOWN and END still waiting to be executed. This process continues until player 6 is called, follows the first command and sees 5 WOWS standing and stops the procedure telling player 5, "I'm done." Player 5 looks at the list of unfinished commands, sits down, and tells Player 4, "I'm done." This process continues until each player in turn finishes the previously unfinished instructions and informs the player who had originally called her that she is done. Player 1 is the last to finish, and tells me that she is done, returning the command to top level as I started the whole play off by telling player 1 to WOW3.

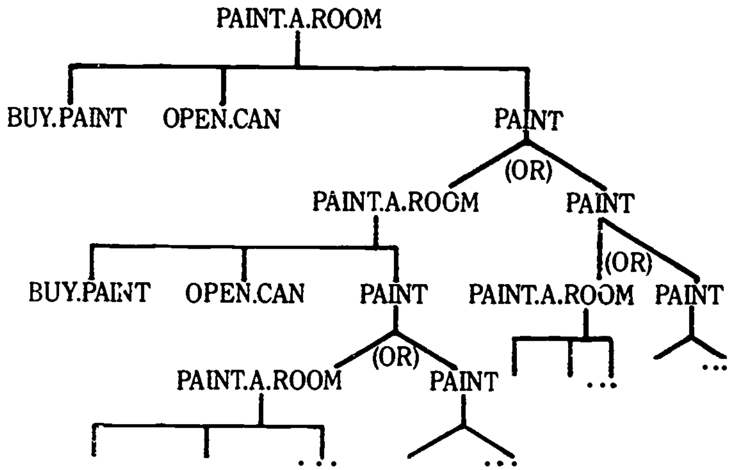
A computer procedure script illustrating embedded recursion is MYSTERY.

```
TO MYSTERY :NUMBER
     IF :NUMBER = 0 STOP
     MYSTERY :NUMBER-1
     PRINT :NUMBER
END
```

The mystery of this classic Logo procedure is answering the question about what the procedure does (Abelson, 1982). Once again I ask teachers for predictions about what they think will happen before they perform the play. The WOW example of embedded recursion will have informed their prediction. Then by actually performing this procedure as a play, an image for how embedded recursion works will become clear and the internal logic of Logo as a programming language will be seen.

Now that the basic idea of recursion is clear, it is time to move into another level of complexity with an example of mutual recursion. In this form of recursion, the recursion line in a procedure calls the procedure which originally called it, rather than calling itself. The clearest example of this I've been able to think up uses a new version of ZEEP, a Logo metaphor story originally created by Danny Hillis at MIT in the 1970s (Watt and Watt, 1985). The story is a people procedure play.

ZEEP is a friend from Mars who doesn't know anything about living on earth. Although she is very smart in math calculations, we have to teach her everything else. One day she decides to paint her room so we have to write a superprocedure to teach her how to do the job.




```
TO PAINT.A.ROOM
BUY.PAINT
OPEN.CAN
PAINT
END

TO BUY.PAINT
WALK TO STORE
ASK THE PERSON FOR A CAN OF PAINT
PAY FOR IT
CARRY IT HOME
END

TO PAINT
IF THE CAN IS EMPTY PAINT.A.ROOM STOP
OTHERWISE DIP THE HAIRY END OF BRUSH INTO PAINT
RUN THE PAINT FROM THE BRUSH ONTO THE WALL
PAINT
END
```

You will immediately recognize the richness of teaching concepts in this story. (top down programming, procedures calling other procedures, trees, one procedure does one job, debugging, identifying primitives, stop rules, etc.) I am using this verison of the ZEEP story in this article for one teaching idea, to demonstrate how recursion can be part of a superprocedure. In this example, PAINT uses tail recursion to keep painting until the can is empty at which point it calls PAINT.A.ROOM which calls first BUY.PAINT, then OPEN.CAN, and then PAINT again. This is an indirect mutual recursion.

COMPUTERS IN THE CLASSROOM

This article demonstrates some ways I involve the teacher/learners I teach to continue to use their bodies to learn to think Logo (M. Watt, 1984 b). These plays provide a focus for class discussions and demonstrations about the ideas embedded in the Logo language. They provide social interaction, a learning context often chosen by women. Putting on plays allows a lighthearted approach and a playful manner while building some powerful imagery about recursion and its Logo syntax.

Note: These ideas were developed in collaboration with Dan Watt and many students. Conversations with Tony Stavely, Tim Riordin and Ramon Zamora furthered my thinking on this subject. Some of the references not specifically cited in this paper contributed to my own deepening understanding of the process of recursion.

Molly Watt is an educator with twenty years' experience as a teacher and administrator in both public and private schools. During the past four years, she has devoted most of her professional work to learning about educational computing and supporting teachers, administrators, children, and parents in learning along with her. You may have seen her "Ask Molly" column in ***Teaching and Computing Magazine*** *or some of her other published work in popular magazines and journals. Her interests beyond LOGO include software evaluation and the computer as a tool in the writing class. For more information on her work, contact:*

Molly Watt
Educational Alternatives
Gregg Lake Road
Antrim, NH 03440

REFERENCES

Abelson, Harold, 1982. **Logo for the Apple II.** BYTE Books/McGraw-Hill.

Abelson and Sussman, 1985. **Structure and Interpretation of Computer Programs,** M.I.T.

Hofstadter, Douglas, 1979. **Godel, Escher, Bach.** Basic Books.

Papert, Seymour, 1980. **Mindstorms.** Basic Books.

Poundstone, William, 1985. **The Recursive Universe,** William Morrow.

Riordan, Tim, 1984. "Helping Students with Recursion: Teaching Strategies I, II, III," **The Computing Teacher,** December/January, 1983–4, February, 1984, March, 1984

Stavely, Tony, December 1982. "List Processing in Logo," **The Computing Teacher.**

Watt, Daniel, a) **Learning with Logo,** McGraw-Hill, 1983; b) "Logo: What Makes It Exciting?," **Popular Computing,** August, 1983

Watt, Molly, a) "Teaching Recursion By Playing Turtle and Playing Procedure," **AEDS/ECCO PROCEEDINGS,** April, 1985; b) "The Robots Are Coming," **Popular Computing Guide to Computers in Education.** Mid-October, 1984 issue; c) **Welcome to Logo! Workbooks and Teachers' Manuals.** D.C. Heath, 1985.

Watt, Molly and Watt, Daniel, 1985. **Teaching with Logo.** Addison-Wesley.

MATH CLASSROOM PROBLEM SOLVING WITH NOTEBOOK COMPUTERS

by Dawes Potter

*Editors' note: In addition to being lighthearted and fun to read, Dawes Potter's article does us the service of showing how programming on small, inexpensive computers can be placed in the service of mathematics instruction. It's not about learning to program, though one suspects that his students do learn quite a bit of that, but obout learning **through** programming. The style of using computers—as a tool-maker, to borrow Judah Schwartz' term—is an extraordinarily flexible one, and variations on Potter's theme are easily imagined. Given the low cost of the notebook and even smaller computers such as those Potter employs, it should be within the reach of many schools to go beyond imagining variations to putting them into practice.*

"Okay, Class. Let's see you find three consecutive integers, the first divisible by 5, the next by 6, and the last by 7. And NO FAIR answering 5, 6, 7. Come up with a triple that's not trivial. Suggestion: Think a bit about what calculations the problem requires and see if you can come up with an algorithm that will do them for you. Then take out your computers and type in a program that will. . ."

Wait a minute! You don't really mean **take out** your computers, do you? Don't you mean **wheel** them out or **lug** them out?

No, I mean **take** them out—of my briefcase or my desk drawer or my knapsack. We're talking here about 33 computers from six ounces (pocket) to one pound (handheld) to four pounds (notebook). All are self-contained, with their own LCD screens; the one and four pounders have dot-addressable graphics.

After a short fight over who gets the **Great Big** one (the Tandy 200 four pounder, with a foldup 240 x 128 dot screen), or the **big** ones (the Radio Shack 100 and the NEC 8201A, both 240 x 64), the rest of the class settles down with the **regulars** (the Casio PB-700's, 160 x 32), or the **tinies** (the Sharp 1250's).

A little thought on the problem and the kids see that the first number has to end in 5. Zero is out because the second would have to end in 1, no good for dividing by 6. So they start out at 15, and write a program looking something like this:

```
10 FOR N = 15 to 1005 STEP 10
20 IF (N+1)/6 = INT((N+1)/6) THEN IF (N+2)/7 = INT ((N+2)/7
THEN PRINT N;N+1;N+2
30 NEXT N
```

We're all surprised that 215, 216 and 217 is the first triple meeting the conditions. Bigger than we would have guessed. But the computers don't stop there: 215, 425, 635, and 845 all begin triples that work. So instead of just solving the original problem, our program has created several more. How come the first triple begins with 5 + 210 and after that the triples all differ by 210? Isn't that the LCM of 5, 6 and 7? What about quadruples divisible by 5, 6, 7 and 8? Will the first one begin with 845, which is 5 more than the LCM (840) of the four numbers? We add THEN IF (N+3/8 = INT((N+3)/8) to LINE 20, and ask it to PRINT N+3. Our hunch if verified.

Remember, this is NOT a computer class we've been talking about. It is an Integrated Course 2 class of ninth and tenth graders. By hand we might have spent half the period trying to find that 215, 216, 217 triple. With the computers the whole investigation, including creating and typing in the program, took something like 10 or 12 minutes.

It's this on-the-spot problem solving that makes our little computers so valuable. Two students can easily carry all 33 of them from one classroom to another. No plugs or outlets are needed, since they're all battery powered. On the average,a set of four AA batteries lasts the one and four pounders about six months. The six ouncers are still going strong on their original batteries after almost two years. All this means that the **computers go to the students** instead of the other way around. It's usually too much trouble to take your class to the computer room to solve a quick problem or two, even if it's right next door. If it's only a matter of opening your file cabinet or sending two students down the hall to pick them up from a colleague, then it's well worth the time.

Our assumption in September is that the kids don't remember or never did know their BASIC programming. We start from scratch and—of course—encourage them to teach each other as we go along. You may rightfully ask, "How do you have time for computers and still get over the year's work?" By **keeping the programs short**—10 or 12 lines at the most, by **making them relevant to the day's work,** and by **putting a daily time limit on computer programming**. These are not particularly confining rules. Unlike conventional PCs and micros, our battery powered, room-to-room computers remember all their programs even when shut off. And depending on the model, they have from two to 72 K, and from 10 to 40 different storage areas, each one reservable for a different class. No booting up; no disks to lose. Programs from the day before are oven ready.

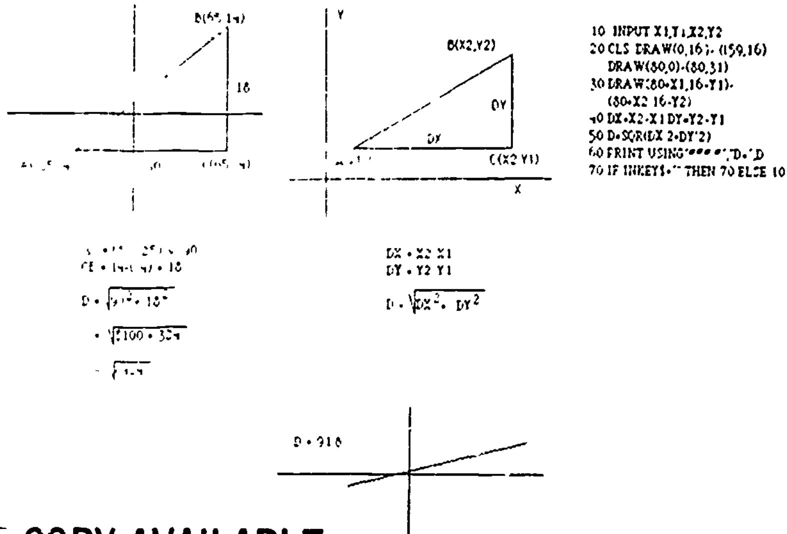
Here's a three-pronged approach that integrates our little computers into the day's work. Suppose we're introducing the formula for the distance between two points A(X1,Y1) and B(X2,Y2). I'll ask three students to go up to the board, with the understanding that the one on the left will be responsible for the **numerical** distance AB, the one in the middle for a **literal** solution, and the one on the right for a **computer** solution.

Leftboard draws a free-style X-Y coordinate system and locates any two points, say A(−25,−4) and B(65,14). Centerboard likewise draws an X-Y axis with points A(X1,Y1) and B(X2,Y2). We then ask, "Rightboard, what are the corresponding lines for a program?" Rightboard replies that we have to **input** the coordinates of the two points A and B, clear the screen, and draw X-Y axes and line AB. So Rightboard writes:

```
10 INPUT X1,Y1,X2,Y2
20 CLS: DRAW (0,16)–(159,16): DRAW (80,0)–(80,31)
30 DRAW (80+X1,16–Y1)–(80+X2,16–Y2)
```

Lines 20 and 30 will vary a bit, depending on which of our computers a student is using, but they will all center the origin at the middle of the LCD screen in line 20 and work from that origin to draw the points in line 30.

"Okay, Leftboard. What's next?" So Leftboard hangs the appropriate right triangle under the line segment AB and finds that the sides DX and DY are 90 and 18. Centerboard finds DX = X2–X1 and DY = Y2–Y1, which Rightboard translates into BASIC as line 40 DX = X2–X1: DY = Y2–Y1. And so it goes, until we have three boards looking something like this:



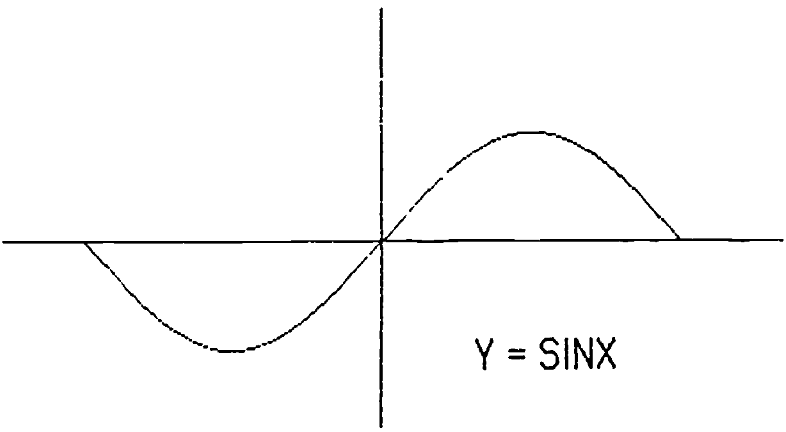
After seeing the problem solved three different ways, and after finding a couple of distances by hand and maybe a dozen more (including the drawings) on their little computers, the students end up with a pretty firm hold on the distance formula. They have the added satisfaction of using a program they designed themselves.

To challenge my students and to improve their graphing techniques I sometimes say, "Put together the shortest program you can (using the fewest bytes) that will draw the graph of any equation we choose on an X-Y axis, with the added requirements that we can select the domain (A to B), the increment of the independent variable (D), the scale of the graph (S), and the location of the origin (H,K) on the LCD screen."

Because we're rewarding **brevity** and **simplicity**, we'll leave out all remarks. It's to be a quickie "throw-away" program, like a paper plate. Out of such an assignment will come something like this:

```
10 CLS: INPUT A,B,D,S,H,K: CLS: DRAW(0,K)-(159,K):
DRAW (H,0)-(H,31): ANGLE 1
20 FOR X=A TO B STEP D
30 Y=SIN(X) or whatever function you choose
40 P=H+S*X:Q=K-S*Y:DRAW(P,Q)
50 NEXT X
60 IF INKEY$="" THEN 60 ELSE 10
```

Maybe there are 100 bytes here. A couple of minutes of typing and we have a program that will do everything except fry eggs. If we take a domain from −pi to pi, with increment .1 radians, scale 14, and center (80,16), we get this picture:

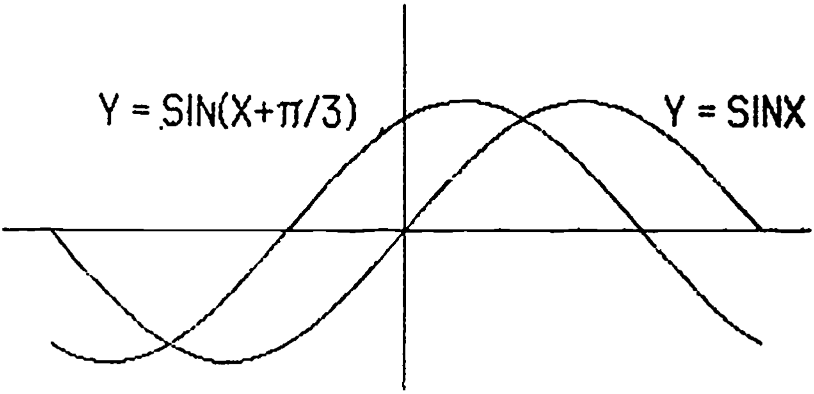


BEST COPY AVAILABLE

Frequently my students think SIN(X+PI/3) should be the **sine** curve moved over to the **right** a bit. Challenge: Change our graphing program to draw this new graph, along with the original, to see what happens. Add these lines:

```
35 Z=SIN(X+PI/3)
36 U=K-S*Z
45 DRAW(P,U)
```

Out comes this pair of curves, showing clearly that SIN(X) moves **to the left** when PI/3 is added to the argument.

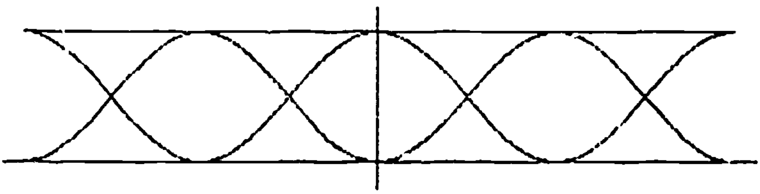


Our "throw-away" program leads to countless problem solving challenges. Modify it, for example, to illustrate **graphically** the Pythagorean trig identity (SIN(X))^2 +(COS(X))^2=1. It isn't long before a student comes up with these new lines:

```
30 Y = (SIN(X))^2: Z=(COS(X))^2: W=Y+Z
35 U=K-S*Z: V=K-S*W
45 DRAW(P,U): DRAW(P,V)
```

Here's the result, a nice verification of an identity most of us have at best checked for a point or two at a time, but never for enough values to draw convincing graphs. This version of our modest little graphing program can easily be modified to verify all kinds of trig identities.

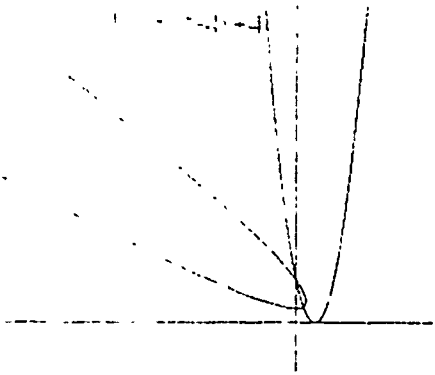
$SIN^2X + COS^2X = 1$



Rotating graphs is another tricky concept which our simple program can help clarify. What changes are needed to show an original graph and a new one rotated through an angle of T radians?

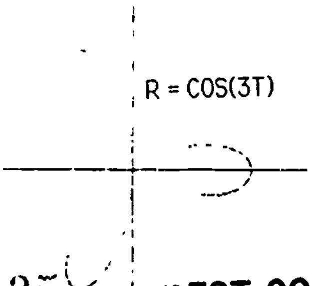
```
 5 INPUT T for the angle of rotation
30 Y=X^2-4*X+4 (Let's draw a parabola this time.)
35 X1=X*COS(T)-Y*SIN(T): Y1=X*SIN(T)+Y*COS(T)
36 U=H+S*X1: V=K-S*Y1
45 DRAW(U,V)
```

A run of the program with T = pi/3 gives a picture like this:

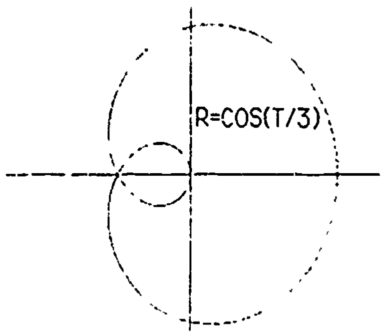


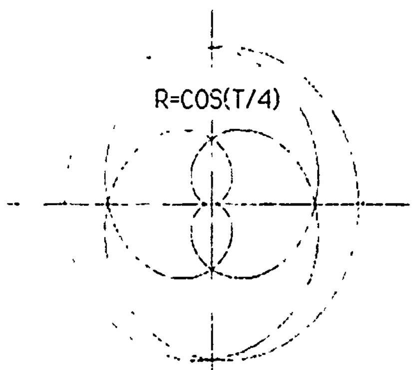
Can this program draw polar coordinate graphs? Easily. All the inputs in the original program can stay the same to select the domain, increment, scale, and location of the origin. We might as well also stay with the X-Y axes instead of using a polar axis. To convert to the customary R for radius and T (theta) for angle, we might do well to rewrite lines 20 through 50 something like this:

```
20 FOR T = A TO B STEP D
30 R=COS(3*T) or whatever function we pick
40 P=H+S*R*COS(T): Q=K-S*R*SIN(T)
50 NEXT T
```



In the good old days we used to spend at least a full period drawing a graph like this one, and another period or two figuring out what happened when we changed the function to COS(4 * X) or COS(5 * X). There was usually not enough time to experiment by hand with cardioids, limacons or other interesting curves. With this little program we put together in five minutes, we can study and find patterns in dozens of polar functions. The students end up with a much broader view of polar coordinate applications then I could show them in the old way. There's never time, for example, to have a look at the striking differences between r=cos(t/3) and r=cos(t/4), shown on the screen dumps below.





This is all fine, some may say, but shouldn't you be teaching more **structured Basic** programming? To which I would have to reply, "In long, involved programs, yes, but in these quickie disposable programs, no." We're concentrating on the art of **problem solving,** not computing. Take this one-liner to find nPr:

```
10 INPUT N,R:T=1:FOR S=N-R+1 TO N:T=S*T: NEXT S: PRINT T
```

No more need to save this program, let alone document it, than to keep an old piece of scratch paper unless, of course, the proof of Fermat's last theorem is on it. Even a seven-liner to find nPr, n! and nCr all at the same time can get nicely unstructured. It just flows:

```
10 INPUT N,R:N1=N:R1=R
20 GOSUB 300:PRINT T;:T1=T
30 R=N:GOSUB 300:PRINT T;
40 N=R1: R=R1:GOSUB 300
50 PRINT T1/T
60 IF INKEY$="" THEN 60 ELSE 10
300 T=1:FOR S=N-R+1 TO N:T=S*T:NEXT S: RETURN
```

An average student who's no whiz at computing can usually see that line 300 is the heart of the program, that line 20 finds nPr, 30 finds n! and 40–50 find nCr.

Enter 6 for n and 2 for r. Out comes

```
30    720    15
```

To make these tiny computers work within the limits of a regular math course, it seems to me a teacher has to be realistic (translate cautious) and on the lookout for a couple of dangers:

Letting computers distract you and your students from the main job of teaching and learning math. Firm time limits, as I suggested above, can take care of this.

Having a box of these little computers around and NOT using them, or underusing them only as sophisticated calculators. Promise yourself and the kids you'll use them **for programming** at least 20 minutes a week. They'll become as natural as a pencil, and far more exciting.

Worried students, and therefore worried parents, want assurances that all regular course material will be covered. Keeping on a conventional schedule, and counting computer questions on quizzes not as heavily as the usual material or as extra credit have, I've found, pretty well taken care of that. The first few months are the most difficult. After everyone sees that the computer assisted course moves right along with all the others, the anxiety disappears.

With some care in these matters, and with a tiny touch (but **not too much**) of proselytizing, you and your school can get a little headstart along the road to the future by giving your students some valuable computer problem solving techniques. The regular use of miniature computers in the classroom has already brought me the reward of alumni (including some who said they hated computers) coming back to tell me how valuable the experience was, both personally and in their college studies. I've not often seen such a modest expenditure—maybe $2000—bring so much to so many students. We have 200 students using these handhelds regularly. That's only $10 a student, even if we throw all the computers over a cliff every June and get new ones in September. Our experience with other models tells us we'll be using these for at least four years before new technology makes them obsolete. $2.50 a year per student for some great firsthand math computer programming experience? That's not bad.

Dawes Potter, who teaches mathematics at Scarsdale High School in New York, currently uses handheld computers with graphics in the classrooms. Starting with sliderules in 1947, Dawes has been introducing a variety of learning tools to all of his mathematics classes. Dawes looks forward to the day when most math and science students and teachers will have truly portable battery powered computers at their fingertips. He has given frequent workshops throughout New York on the Scarsdale mathematics program, which will be presented at the national NCTM conference in April 1986. For further information, contact:

Dawes Potter
Mathematics Department
Scarsdale High School
Scarsdale, NY 10583

ENRICHING MATHEMATICS WITH COMPUTER ACTIVITIES

by Eileen Backofen and Barbara Bussey Ringgold

Editors' note: Like Dawes Potter (see "Math Classroom Problem Solving with Notebook Computers"), Eileen Backofen and Barbara Bussey Ringgold teach mathematics through the use of programming, while doubtless teaching a good deal of programming as a by-product. Going beyond the learning of a few primitives to actually learning how to construct real programs inevitably means programming ***about something,*** *so why not make mathematics the subject matter of programs? Or, as Backofen and Ringgold suggest—coming at the matter from the other side—since you have to understand a process well in order to program it, why not use programming to teach mathematical processes and concepts?*

At George Mason High School in Falls Church, Virginia, mathematics students from general math through calculus have access to twenty-two microcomputers in two labs. Within the past two years we have developed a wide range of activities that make effective use of our computers. We wanted to avoid substituting high priced software for activities more effectively handled by an inexpensive workbook. Therefore, our computer projects and assignments are primarily simple routines or public domain programs. When these are selectively combined with traditional topics, the resulting units are greatly enhanced.

Computers can help students visualize abstract and concrete mathematical ideas. Sitting in a math class and having a concept described is not nearly as effective as seeing it "happen" on the screen. The idea of a limit is a good illustration of this. Defining e as $\lim_{n\to\infty} (1+1/n)^n$ is not as meaningful as watching the computer calculate the value of $(1+1/n)^n$ for increasingly larger values of n.

```
5 REM PROGRAM TO APPROXIMATE E
10 FOR N = 1 TO 100
20 LET E = (1 + 1/N)  N
30 PRINT N,E
40 NEXT N
50 END
```

We introduce limits to our students in a pre-calculus course during a unit on sequences and series. In one successful project, the students discovered that some sequences have limits and tried to draw a conclusion as to what the limit would be. It is very easy to see convergence and divergence when the computer is generating terms.

```
10 REM PROGRAM TO EXAMINE TERMS OF SEQUENCES
20 PRINT"N", "TN"
30 FOR N = 1 TO 20
40 LET TN = (2*N+3)/(N+1)
50 PRINT N,TN
60 NEXT N
70 End
```

The projects themselves required little knowledge of programming, We did, however, include extra programming problems for those with more experience. One involved the terms and summation of Fibonacci sequences.

```
10 REM PROGRAM TO ADD CONSECUTIVE SQUARES
15 REM OF FIBONACCI SEQUENCE
20 REM RESULT WILL BE EVERY OTHER FIBONACCI NUMBER
30 DIM F(14),S(10)
40 LET F(1)=1
50 LET F(2)=1
55 PRINT F(1)" "F(2)" ";
60 FOR N=3 TO 14
70 LET F(N)=F(N-1)+F(N-2)
80 PRINT F(N) " ";
90 NEXT N
95 PRINT : PRINT
100 FOR I=1 TO 10
110 LET S(I)=F(I)^2 + F(I + 1)^2
120 PRINT INT (S(I))" ";
130 NEXT I
```

Another problem required the students to expand $(1 + 1/n)^n$ using the binomial series expansion and take the limit as $n \to \infty$. The result is $1 + 1 + 1/2! + 1/3! + 1/4! \ldots$ Then, they had to write a program to compute the sum of the series for increasingly more terms. This will converge very quickly on e.

```
10 REM PROGRAM TO APPROXIMATE E USING SERIES
20 LET S=1
30 LET P=1
40 FOR X=1 TO 25
50 LET P=P * X
60 LET S=S + 1/P
70 PRINT X,S
80 NEXT X
```

Easier concepts also can be visualized with the computer. Graphing and finding roots of a polynomial function is a good example. We adapted a program that graphed given functions, determined integral roots, and also identified what consecutive integers the other roots were between.' The students ran the program with various functions. From the information they received, they determined the number of real and complex roots and as many of the non-integral roots as possible. This exercise is excellent reinforcement for the concepts of degree of a polynomial, the graphic significance of a root, and the difference between complex and real roots.

Some of our most successful computer activities encourage students to develop problem solving skills. They discover that computers don't work quite the way we do but must be given step-by-step instructions. Giving students a task to accomplish with the computer can be fun and thought provoking. A few examples will help illustrate this.

Most versions of Basic have a greatest integer function (INT). In one algebra II class, about half the students had some programming experience in Basic. After a discussion on scientific notation, the students were placed in groups of two or three (with at least one "programmer" in each group) and given instructions to figure out how to get the computer to round to the ones place. (Add .5 to the number and take the INT.) The next assignment was to get the computer to round a number to any place. If the students entered a number such as 34579 and a place value such as 100, the printed result would be 34600. (A little more thought was required—divide by the place, round, multiply by the place.) The final assignment was to have the students get the computer to round to one significant digit and write the number in scientific notation. (A loop was required. They had to keep dividing the number by 10 until the result was a one-digit number and keep track of the number of times they divided.)

```
10 REM PROGRAM TO ROUND POSITIVE NUMBERS
15 REM TO ONE SIGNIFICANT DIGIT
20 REM AND WRITE IN SCIENTIFIC NOTATION
30 HOME
40 PRINT "GIVE A NUMBER"
45 LET C = 0
50 INPUT A
55 PRINT A" = ";
60 IF A / 10 < = 1 THEN 100
70 LET A = A / 10
80 LET C = C + 1
90 GOTO 60
100 Let N = INT (A + .5)
110 PRINT N"*10 "C
120 PRINT "IF YOU WISH TO TRY ANOTHER TYPE '1'";
```

```
125 PRINT "IF FINISHED TYPE '2' "
130 INPUT D
140 IF D = 1 THEN 30
150 END
```

A follow-up assignment could have the students round two numbers to one significant digit, write them in scientific notation, and then find the product.

A good, short algebra I project involved using Apple high resolution graphics. After a unit of graphing and finding the intersection of lines, the students were given five lines to graph on regular graph paper and told the result should look like a star. They were then to find the points that correspond to the five corners of the star and transfer this star, exactly, to the high resolution screen. A problem arose because the computer screen is a vertical reflection of the first quadrant. The students were told the new origin but had to translate the other points. Although this was not a difficult assignment, the result was an interesting review lesson and an introduction to the idea of translating graphs.

Computer experience for lower ability or underachieving students need not be limited to CAI software or drill and practice with bells and whistles. Many topics in general/practical math and pre-algebra can be enlivened and enriched with non-CAI computer activities. Using the computer's capability to generate lots of numbers very quickly we can develop the students' ability to analyze results, look for patterns, make generalizations, and draw conclusions.

For example, as one of the applications of percent, students are taught the concepts of simple and compound interest. The effects of a slightly higher interest rate or more frequent compounding is often lost on students who are struggling with repeated multiplication of fractions and decimals. Using a simple computer program, our students watched a principal of $5,000 grow to over $1,000,000 within a lifetime.

```
5 REM COMPOUND INTEREST PROGRAM
10 PRINT "ENTER THE PRINCIPAL"
20 INPUT P
30 PRINT
40 PRINT "ENTER THE RATE (AS A DECIMAL)"
50 INPUT R
60 PRINT
65 PRINT "ENTER THE TIME (YEARS—IN DECIMAL FORM)"
66 INPUT T
70 PRINT "ENTER # OF INTEREST PERIODS PER YEAR"
80 INPUT N
PRINT : PRINT
```

```
90 FOR K=1 TO N*T
100 I=P*R*(1/N)
110 P=P+I
112 PT=INT (100*P+.5)/100
115 PRINT "AFTER";K;"PAYMENTS YOU HAVE"; "$"; PT
120 NEXT K
125 PRINT : PRINT
130 GOTO 10
```

To convince the skeptics who doubted this result, we checked the first three or four interest payments by hand to reassure ourselves that the computer was applying the formula correctly. The students also saw the necessity of inputting correct data and accurately converting percents and units of time to the decimal form required by the computer.

An extension of this topic was a project where each student searched real estate advertisements for three houses in various price ranges. As might be done in a realtor's office the students discussed the amount of down payment required, current interest rates and computed the amount they would need to borrow. They then entered the data into a home mortgage program[2] and observed the results. They were very surprised that the amount to be repaid over the life of the mortgage is often three to four times the cost of the house. The difference in total repayment amounts and monthly payments among 20, 25 and 30 year mortgages can be quickly computed by rerunning the program with new input. Finally, we discussed the yearly salary required to obtain a given mortgage and the students reevaluated their ability to buy their dream houses.

The use of an enjoyable computer activity can give a new twist to topics often considered boring by students. The metric system is one example. The students had to imagine themselves traveling abroad and responding to quantitative questions with metric estimates, e.g. "What is your height and weight?" These metric responses were entered into a simple conversion program, and each student received a printout of his answers converted to English units. At first we had a large supply of giants and Lilliputians. The program was rerun and new questions added as they became more familiar with metrics.

Students are often fascinated by very large numbers. We built on this interest during the unit on divisibility and prime numbers. The students entered and ran a program which lists the factors of any number input. Then, using their knowledge of divisibility rules to make intelligent guesses, they tried to find the largest prime number they could.

```
10 REM THIS PROGRAM DETERMINES FACTORS (DIVISORS)
   REM OF NUMBERS ENTERED
```

```
20 PRINT
30 PRINT "INPUT A NUMBER"
40 INPUT A
50 PRINT "THE FACTORS (DIVISORS) OF ";A;" ARE:"
60 FOR B =1 TO SQR (A)
70 IF A/B < > INT (A/B) THEN 110
90 PRINT B, A/B
110 NEXT B
130 PRINT
160 PRINT
190 GOTO 20
```

A traditional introduction to probability has been to have each student flip a coin or toss a die a certain number of times. This activity took on an added dimension when the "flipping and tossing" was done by computer.

```
10 REM COIN TOSSING PROGRAM
20 H=0
30 T=0
40 PRINT "HOW MANY TIMES SHOULD I FLIP THE COIN?"
50 INPUT T
60 FOR I=1 TO T
70 R=INT (2*RND(1)+1)
80 IF R=1 THEN 110
90 PRINT "TAILS ";
100 GO TO 130
110 PRINT "HEADS ";
120 H=H+1
130 NEXT I
160 PRINT
150 PRINT "THE RESULT WAS HEADS";H;" TIMES"
160 PRINT "                    AND TAILS";T-H;" TIMES"
170 PRINT : PRINT
180 GOTO 10
```

A much larger number of trials was possible; several coins or dice could be involved; results were seen more quickly and could even be shown graphically.[3]

We realize that our activities were initially well received by our mathematics students because of the natural attraction of adolescents to computers. As they proceeded, however, students came to appreciate the freedom to concentrate on concepts rather than tedious repetitive computation. They discovered they could use their creative ability to direct and control a powerful machine. The resulting improvement in self-esteem was an unanticipated benefit. We feel the contribution of computers in the classroom is limited only by our imagination.

REFERENCES

1. M. Dolciani. 1982. **Algebra and Trigonometry Structure and Method**, Book 2, Boston: Houghton Mifflin, p. 310.
2. K. Armstrong. 1981. "Home Mortgages," San Mateo Office of Education.
3. T. W. Shilgalis. November 1982. "Probability," Illinois State University.

Eileen Backofen and Barbara Bussey Ringgold are co-chairpersons of the mathematics department at George Mason Junior/Senior High School in Falls Church, Virginia. Both teach mathematics, computer science, and computer literacy to students in grades 7-12, as well as sponsoring the Math Team and Computer Club. For further information on their program for enriching mathematics with computer activities, contact:

Eileen Backofen and Barbara Bussey Ringgold
2668 Peterborough Street
Herndon, Va 22071

WHAT WOULD HAPPEN IF. . .?

by Theresa M. Reardon

Editors' note: Theresa Reardon's is one of a series of articles about teaching mathematics through programming. Reardon stresses the general advantage of this approach—the shift in emphasis from computation to concepts. In this context, her article is useful not only for the specific techniques she describes, but as an example of using the computer to facilitate mathematical exploration as a way of developing an understanding of fundamental concepts and relationships.

Computer technology has the power to reshape the fundamental methods of doing and teaching mathematics (NCTM 1985). Traditional methods using standard textbooks often rely on the development of tricks and arithmetic procedures in the explanation of the mathematics (House, Wallace, Johnson 1983).

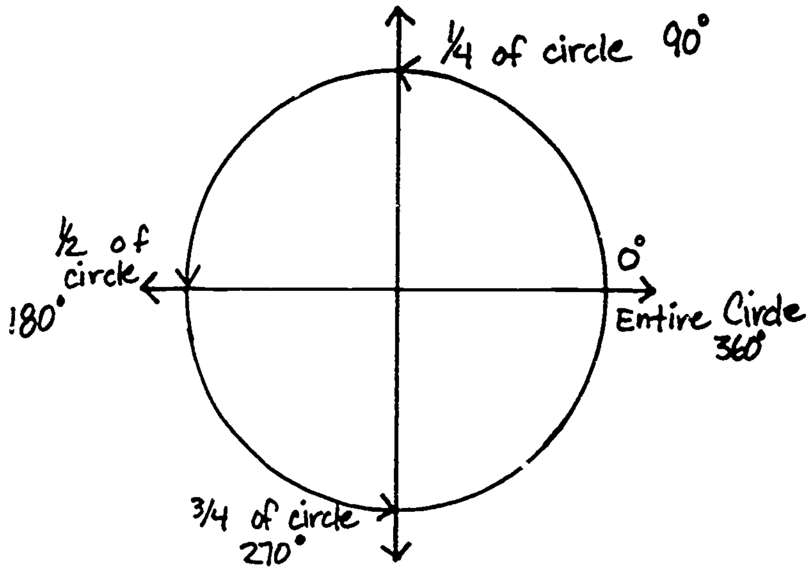
"What would happen if. . .?" is a question many teachers would like to pose to their students. Unfortunately, the very topics we would like them to explore often require complicated computations and a thorough understanding of the concept in order to pursue the exploration, defeating the purpose of the exploration in the first place. Incorporating available computer technology into regular mathematics classes can shift the emphasis from algebraic manipulations and computational skills to an emphasis on developing concepts, relationships, algorithmic design and problem solving (NCTM 1985).

Trigonometry, particularly the topic of circular functions, is one area which fits this description. This topic is traditionally approached from the standpoint of graphing; which first requires students to possess an understanding of the concepts in order to begin the process of graphing and second, that students possess sufficient skill in graphing in order they might evaluate the results. One student, when asked how the value 2, in the equation $f(x)=2\cos x$, affected the shape of the graph, replied, "It makes it twice as tall." This same student, though, could not draw an accurate graph of this function due to confusion about how to label the axes. The student understood that "twice as tall" meant a comparison to some other graph (i.e., $f(x)=\cos x$) but how one labels the axes depends upon the interval used and not the specific equation. As in this example, students have ample opportunities to miss the point of the instruction, which is, that they be able to recognize the relationship between different circular functions and are able to explore the effects of different variable substitutions which lead them to an understanding of the concepts of amplitude, period and phase shift. They cannot see the forest for the trees.

A slightly different approach might help the situation. Most students, by the time they reach the topic of circular functions, have had experience with circles. They have used a compass in geometry to draw circles and in algebra they have used the equations of circles to graph them. It is natural, then, to begin the study of circular functions with the study of circles. The following is an Applesoft **BASIC** computer program for Apple computers which draws circles. Its development was prompted by the desire of several seventh grade students to find a way to draw "fast circles." The program they had been using involved square roots and was, in their words, "gross." This program involves parametric equations, which are not necessary, but make the progam more familiar to students.

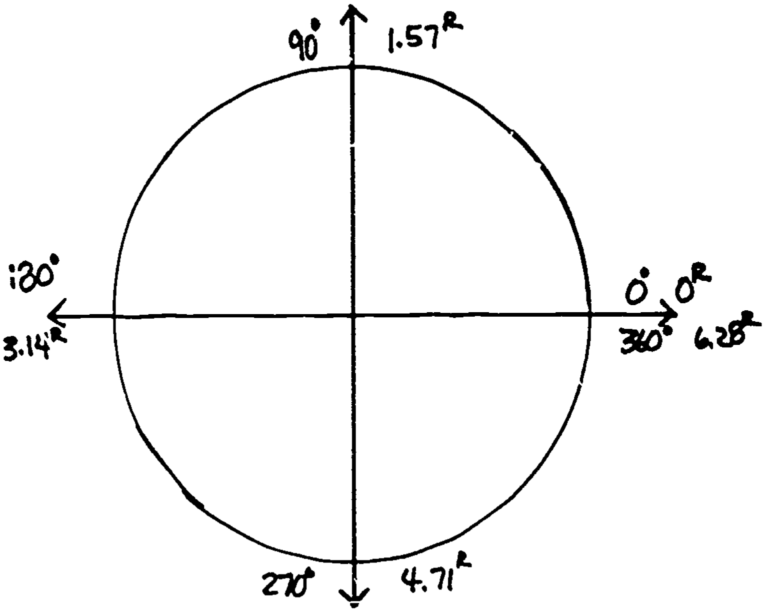
```
5  REM ************************************
10 REM *THIS PROGRAM DRAWS CIRCLES OR ELLIPSES IN HIGH *
15 REM *RESOLUTION GRAPHICS OR HGR. THE VARIABLE R1    *
20 REM *IS THE HORIZONTAL RADIUS AND THE VARIABLE R2   *
25 REM *IS THE VERTICAL RADIUS. FOR A CIRCLE THEY      *
30 REM *WOULD BE THE SAME NUMBER, FOR AN ELLIPSE, THEY *
35 REM *WOULD BE DIFFERENT NUMBERS.                    *
40 REM ************************************
45 HOME
50 HGR
55 HCOLOR=3
60 REM DRAW AXIS
70 HPLOT 140,0 TO 140, 159
75 HPLOT 0, 80 TO 279, 80
80 REM INPUT RADIUS
90 VTAB 21: INPUT "WHAT IS YOUR HORIZONTAL RADIUS"; R1
100 VTAB 23: INPUT "WHAT IS YOUR VERTICAL RADIUS"; R2
105 REM CIRCLE ROUTINE
110 REM
120 FOR T=0 TO 6.28 STEP 6.28/100
130 REM CHOOSE ORDERED PAIRS: PARAMETRIC EQUATIONS
135 X=R1*COS(T)+140
140 Y=R2*SIN(T)+80
145 REM PLOT ORDERED PAIRS
150 HPLOT X,Y
155 NEXT T
160 REM TRY AGAIN?
165 HOME
170 VTAB 21: INPUT "AGAIN"; D$
180 IF D$="Y" OR D$="YES" THEN 45
185 END
```

This program is very adaptable. It has been used with all levels of secondary students (grades 7-12) and does not require extensive programming skills but does require some knowledge of Applesoft graphics. Specifically, Apple computers use radian measure rather than degree measure in computation and locate the origin at the upper left corner of the screen rather than the lower left or center. The first problem is easily overcome by presenting radian measure to students as an alternative to degree measure. At this point it is helpful to draw a circle on a coordinate axis with the center at the origin and pose the questions, "Through how many degrees would I pass if I were to go all the way around the circle? If I were to go only half way around? If I were to go one-quarter or three-quarters of the way around?"



By asking the students to label a few key degree measures such as 0, 90, 180, 270 and 360 in this way, it reinforces the notion that degree measure is directly related to specific distance on the circle's circumference.

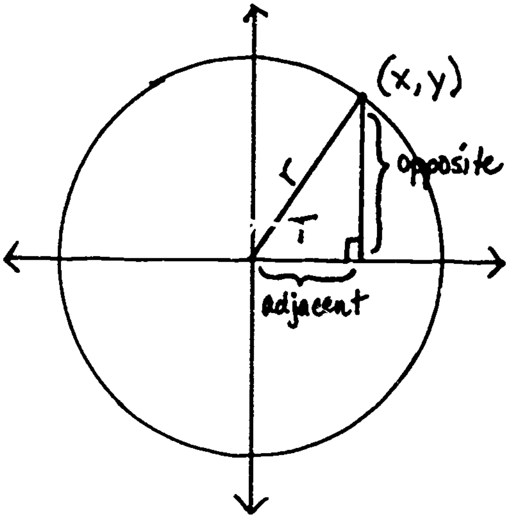
Some may remember from geometry that the circumference of a circle is 2π times the radius or $c=2\pi r$. If we let this circle have a radius of one unit, the circumference is 2π units. If we substitute 3.14 for π, the circumference is now 6.28 units or radians instead of 360 degrees. Again, ask the students to compute a radian measure for the corresponding degree measure in order that they may relate this new measure to one with which they are familiar.



The second problem, the origin location, can be presented as a way for students to draw the circle anywhere they would like within the limits of the graphics screen. The screen extends horizontally from 0 to 279 and vertically from 0 to 159. If we wanted to locate the circle in the center of the screen, for example, we would need to add to each x-coordinate 140 (half the horizontal distance) and to each y-coordinate 80 (half the vertical distance) to translate the "old" origin to the "new" origin. This has been done to the parametric equations in lines 135 and 140. One activity to illustrate this capability would be to change the program to draw circles in many different locations by changing the values added to the x,y-coordinates in lines 135 and 140.

The use of parametic equations to graph a circle is very effective. The circle can be thought of as a series of ordered pairs (x,y). At this level, students should be familiar with right triangles and with the Pythagorean theorem. The parametric equations can be developed through the use of a right triangle as in the following example:

> Draw a right triangle with one vertex at the origin and one vertex on the circle. The vertex on the circle is the point (x,y). Call the angle the radius makes with the x-axis T.



In right triangle trigonometry, the cos (T) is defined as the ratio of the adjacent side to the hypotenuse (radius of the circle) or cos (T)=x/r. The sin (T) is defined as the ratio of the opposite side to the hypotenuse or sin (T)=y/r. This leads directly to the parametric equations:

rcos(T)=x   rsin(T)=y

and implies that every point on a circle can be written as (rcos(T), rsin(T)) where r is the radius of the circle and T is the measure of the angle the radius makes with the x-axis at that point. This angle can be measured in degrees or in radians. Beginning with T=0, we can graph many points, using a loop to generate the individual points. The number of points can be controlled by the STEP used. In this case, we are graphing 100 points or STEP 6.28/100.

This program has several applications. Students can use this program to explore various circles and to experiment with size and shape. They are able to vary the horizontal and vertical parameters as well as location on the screen. This aspect of the program is valuable when discussing the necessary and sufficient conditions for a figure to be a circle or ellipse. Younger students, especially, need this experience. By using appropriate choices for R1 and R2 in lines 135 and 140 students can compare circles and ellipses and draw conclusions leading to generalizations about the relationship between them. Teachers can facilitate the discussion by asking questions such as, "What do I need to change in the program to generate a circle of size. . .? An ellipse? Which variables control location and which control size or shape?" Also, we are able to generate any regular polygon by changing the number of points plotted. This can be done by adding the following lines to the program:

```
115 INPUT "HOW MANY POINTS"; N
116 HPLOT R1+140,80
117 HPLOT TO R1+140,80
```

and change the following lines:

```
120 FOR T=0 TO 6.28 STEP 6.28/N
150 HPLOT TO X,Y
```

At this level, students become familiar with the circle as a series of plotted points which can be manipulated to produce various figures.

This basic program can be modified to graph circular functions. If we define the cosine function as one in which a distance along the unit circle is mapped to the first coordinate of its corresponding ordered pair, it is easy for students to form a mental image of the process given their earlier work with circles. They understand that this distance is the loop from T=0 to T=6.28 and that ordered pairs are formed by substituting these values into the parametric equations in lines 135 and 140.

Again, Applesoft graphics is somewhat at odds with traditional graphing. If we watch the circle program long enough we will notice that the circle is graphed in a clockwise direction. This is the opposite direction from which we develop the notion of angle measure and distance on the unit circle. To ensure that the function will be graphed in the familiar manner, R1 must be multiplied by -1 (line 135). Since the graphed points are very small, it is also necessary to multiply T by 10 so that the graph can be seen easily (line 150). As previously described, the horizontal and vertical components must be adjusted so the "center" of the circle is the center of the screen. The **BASIC** program can be modified by changing the following lines:

```
80 REM INPUT AMPLITUDE
90 VTAB 21: INPUT "WHAT IS YOUR AMPLITUDE"; R1
135 X=-R1 * COS(T)+80
[illegible] HPLOT 10*T+140, X
```

and deleting the following lines:

```
100 VTAB 23: INPUT "WHAT IS YOUR VERTICAL RADIUS"; R2
140 Y=R2*SIN(T)+80
```

The result will be the familiar graph of the cosine function. It is a valuable experience for students to develop this last modification on their own. When a class of junior high students was presented with the circle program and the question, "How would we graph the cosine function using the mapping definition?" the students began by going back to the parametric equations and trying to figure out how the ordered pairs were generated. One eighth grade student suggested that "we don't need both x and y (as written) because part of our ordered pair will be T." This observation was an important step in the process of understanding the implications of the statement, "the cosine function is a mapping of a distance on the unit circle to the first coordinate of its corresponding ordered pair." Many students never realize the relationship between the points on the unit circle and the points on the cosine graph.

From this point, students can make other adjustments in the program to graph sine, tangent, secant and cosecant functions using the relationships developed from their work with triangles. For example, to graph the tangent function make the following changes in the program:

```
135 X=-R1*TAN(T)+80
```

and add the following line to test for illegal values:

```
140 IF X > 159 OR X < 0 THEN GOTO 155
```

Students also can adjust the interval over which the functions are graphed by changing the parameters for T. This allows students to graph different parts of a function and to compare graphs of functions for similarities and differences.

The ability to use a computer to do the actual graphing removes the algebraic manipulative aspect from the study of circular functions and gives both the student and the teacher freedom to explore. Developing the actual **BASIC** program forces students' understanding of the concepts involved. They have the option to pose such questions as, "What would happen if we multiplied T by something...?, or "What would happen if we graphed a circle using these parametric equations: X=R1*COS(5*T); Y=R2*SIN(7*T)...?" Both of these questions lead to concepts involving ideas about phase shift and polar coordinate graphing. Other applications include using the computer to graph the sum or difference of two equations and to explore the sum and difference identities.

Yes, we have "lost something" by using the computer. Students will not have to know how to read a trig table. Students will not have to spend hours learning how to do the physical manipulations involved with graphing. Students will not have to wait until they are in an advanced algebra class in order to learn about other ways to write equations of circles. Students will be able to graph faster and with more style by using computer programs which they have written. Yes, we will have to change the way we teach and what we teach. One of the most immediate and important consequences of computer technology is the need to make informed decisions about certain aspects of the mathematics curriculum (NCTM 1985). It decreases the need for many computational procedures now taught, and in doing so, allows students to easily pose the question, "What would happen if. . .?

Theresa Reardon has taught mathematics and computer programming at all levels. Currently she is the computer coordinator at Mounds Park Academy, a K-12 private school in St. Paul, Minnesota. As a master teacher at the Twin City Institute for Talented Youth she helped other teachers learn to use the graphic capabilities of a computer to teach mathematics. For further information on her use of computer applications, contact:

Theresa Reardon
St. Olaf College
Northfield, MN 55057

REFERENCES

Corbitt, Mary Kay, ed. April 1985. "The Impact of Computing Technology on School Mathematics: Report of an NCTM Conference." **Mathematics Teacher.** Vol. 78, No. 4: 243-250.

House, Peggy A., Martha Wallace, and Mary Johnson. 1983. "Problem Solving as a Focus: How? When? Whose Responsibility?" in **The Agenda in Action 1983 Yearbook of the National Council of Teachers of Mathematics,** Gwen Shufelt, ed. Reston, VA.: NCTM.

Kimberling, Clark. January 1985. "Circles and Star Polygons." **Mathematics Teacher.** Vol. 78, No. 1: 46-51.

——— March 1985. "Graph Many Functions, Part I." **Mathematics Teacher.** Vol. 78, No. 3:195-200.

NCTM 1985. see Corbitt, Mary Kay, ed.

Wagner, William J. April 1982. "Circles, Polar Graphs, and a Computer—Some Unexpected Results." **Mathematics Teacher.** Vol. 75, No. 4: 323-328.

Zalewski, Donald L., ed. 1982. **Microcomputers for Teachers with Application to Mathematics and Science,** School Science and Mathematics Association Topics for Teachers Series No. 3. Bowling Green, OH: SSMA.

CHOOSING A WRITING ADVENTURE

by Anne Weber Dunham

Editors' note: Teaching writing with the use of word processing software—the subject of the articles in another section of this volume—is becoming more and more common. But teaching writing through **programming**? *That's what Anne Webber Dunham describes here, combining the current popularity of "choose your own adventure" books with the appeal of computers to interest previously apathetic students in writing, editing, and programming all at the same time.*

Have you ever felt that trying to get your students interested in writing was as frightening as a jungle safari or a mission as a double agent? Well, with this successful writing and programming assignment, you are free to **Choose Your Own Adventure** and survive to tell about it.

The seventh graders I taught were much like everyone else's students, I suppose. Some were turned on by adventure stories such as Ian Fleming's James Bond. Some were turned on by Garfield, and some were turned on by the cute guy in second period. Few were turned on by English writing assignments. I was as unhappy reading daily journals as they were writing them. "Prince," the popular rock-n-roll singer, and Friday's soccer game tied for first as the favorite topics for student assignments. However, I could only read so many times that "Prince is bad." The school year was only three months old, and students were already beginning to lose interest. My quest became to find a writing assignment that would excite all of my 55 seventh graders, not just the slow learners, and not just the fast learners.

I began to notice that many of my students read **Choose Your Own Adventure** books by the dozen, easy-to-read adventure stories in which the reader takes an active part. He or she turns to and reads certain pages of the book based on his or her decisions as a detective, a knight in shining armor, or some other hero. For example, the end of a page in a detective story might read, "Turn to page 68 if you want to question Mr. Smith. Continue if you choose not to." I had not taken much stock in these books because, even though the students enjoyed them, the required book reviews on them were difficult to write, and they did not seem to challenge my students' reading abilities.

Then, in December I was reading my **Classroom Computer Learning** magazine. In it was an idea by a teacher who had taught her students the use of the INPUT statement by having them write stories with two different endings and then putting them on the computer. My students had been learning **BASIC** programming in English since October. They were beginning to learn the INPUT statement around this time. The wheels began to turn. Couldn't I capitalize on their interest in computers and combine that interest with original **Choose Your Own Adventure** stories? They could learn editing, branching, INPUT, short story writing, and possibly become interested in writing. I began to get excited about the endless possibilities.

Just before Christmas break, I mentioned the idea of writing **Choose Your Own Adventure** stories to my students. They could barely contain their excitement when they returned to school in January. They asked me every day when we could get started. Of course, they wanted to jump right in, but I knew there needed to be preparing, pre-thinking and pre-writing if I wanted good finished products.

I began pre-writing activities with a **Choose Your Own Adventure** entitled **Your Code Name is Jonah.** As I read the book out loud, I let the class decide democratically what direction the hero would take. Avid **Choose Your Own Adventure** readers, knowing their books well, tried to convince other students to make choices that would prevent them from being killed. Getting quite upset when the hero died, these students wanted me to go back and continue on the correct path. I followed the reading with a discussion of how a **Choose Your Own Adventure** is constructed.

Next, my students read Stockton's **Lady or the Tiger?** They were asked to finish the story — a common writing assignment. However, we discussed as a class the mood of the story and how it was conveyed by word choice. My students also became aware of the many possibilities for plot development. "Couldn't the princess jump in the ring and be devoured by the tiger, too?" "Doesn't she carry a hand gun and shoot the tiger when she realizes she made a mistake?" Some students were so caught up in the story they became angry at having to make a choice. They were obviously getting turned on by writing and telling a good story. Ninety-five percent of the students handed in this assignment. (Quite a surprise, since half of my students regularly missed completing homework assignments.) I believed they were ready to begin to **Choose Your Own Adventures.**

Our comput- lab contains only ten computers, so I asked the students to form groups of three or four. Since we hadn't done much group work, I was hoping they would feel more comfortable if they worked with friends. Interestingly, this was a lesson in itself. They realized as they went through the project that their friends were not always easy to work with, or very responsible about doing their share. Their frustrations surfaced when I read their journals and listened to their conversations. A typical journal entry from that time was, "Don't ever put me with Chad again. He won't let me do anything. He thinks all my ideas are stupid and his are great." To which my response would be, "You chose to work with -d," and somehow the friendship continued.

Students discovered their ideas for plots needed to be compromised and agreed upon. For example, one group of girls was working on a romance in which the heroine needed to make a choice about her prom date. Unfortunately, the choice was between two different boys, and one of the favorite high school football players was not chosen. Many students learned that they were the one person in their group who had to do more writing or work at home in order to get the assignment done on time. This caused an argument from time to time. Things were usually resolved quickly, and everyone survived without a scratch.

After spending one period discussing what their story would be about, the first pre-writing assignment was to design a short plot summary. They had to define who the reader would be (for example, a detective) and give two directions the plot could take. This is called branching. When programming a computer, a branch is a place the program takes a different direction based on choices the user makes. I asked the students to write one branch for the reader within a three-page story. Protests abounded. "But what if we want to write more pages?" "We want to write four of five branches and have them choose at the end of every page!" I was surprised! My students actually wanted to do **more** writing! We finally settled on a maximum of ten pages with six different places where their reader could choose a different plot direction.

The next step was a rough draft. I asked the students to double space for revision purposes. All but two of my 55 students completed the entire assignment. During the first editing session, each person in the group was to read the story out loud once. They paid particular attention to punctuation, organization, and spelling. Reading papers orally helped my students to understand how their writing really flowed and where they needed more or less punctuation. Since they had always been hesitant about reviewing their work to correct spelling errors, they were allowed to spread the work among their group. With each student using a dictionary, they didn't mind as much, and it proved to be quite effective.

The students were now ready for their second draft. After writing it, they spent their editing time examining paragraph structure and word choice. We happened to be working on a spelling unit which was based around words used to enhance narratives such as shriek, moan, or grimace. We discussed how they could place their readers in the story more effectively by using these words in place of "said." I also forbade them to use constructions such as "and next," "and then," or "also," because these transitions encouraged run-on sentences for my students. When the draft was completed and revised, they received credit.

After revision, they recopied their final drafts. I was amazed at the quality of these papers. Students who had always turned in two to three page papers laced with spelling and punctuation errors had completed six to eight page papers with minimal errors. I thoroughly enjoyed reading all my students' papers for the first time. Titles ranged from "Quest for the Diamond of Life" to "A Search for Garfield" to "The Case of Ms. Weber's Stolen Candy Jar." To illustrate the level of work and imagination, following are some short paragraphs from these stories.

> As you enter the cavern you immediately notice one thing in particular. Somebody has magically lit up the cave. You know you are not alone. You see a shadow at the end of a narrow cavern.

> Your name is Detective Bird, and you are hired to find Garfield who is supposedly catnapped. He was last seen on the corner of Skills Street wearing a red and blue baseball cap with a G on it. He has been missing since Saturday at lunch time.

> At lunch Nate picked the lock and stole Ms. Weber's candy jar. Ms. Weber called you in to look for her two cent candy and her five cent candy jar that she bought at the D and I. You set out on the quest for the person who stole Ms. Weber's candy jar.

Even though the stories were wonderful, the project was not finished. The students still had to put the stories on computer disks. We spent one day in class discussing how to use **BASIC** programming to save the stories. We had to use the print statement since we didn't have a word processing program. The problem with the PRINT statement is that quotation marks are part of the command. It is necessary, therefore, to use apostrophes in place of quotes. For example, one of the above texts, programmed, would appear as follows:

```
1000 PRINT "WHERE'S GARFIELD?"
1010 PRINT "      YOUR NAME IS DETECTIVE"
1020 PRINT "BIRD AND YOU ARE HIRED TO FIND"
1030 PRINT "GARFIELD WHO IS SUPPOSEDLY CATNAPPED."
1040 PRINT "HE WAS LAST SEEN ON THE CORNER OF"
1050 PRINT "SKILLS STREET WEARING A RED AND BLUE"
1060 PRINT "BASEBALL CAP WITH A G ON IT. HE HAS"
1070 PRINT "BEEN MISSING SINCE SATURDAY AT LUNCH"
1080 PRINT "TIME."
1090 PRINT "      'WHY DON'T WE START TO LOOK ON"
1100 PRINT "SKILLS STREET?' YOU ASK YOUR PARTNER."
```

It took a while for the students to get used to typing like this, but since they got to work on the computers for this assignment, they didn't mind redoing their stories if they made an error.

Next, the students learned the care and use of floppy disks, and since the disks contained their own stories, they made good use of these lessons. I also chose to teach the INPUT statement and looping before they actually began putting their stories on disks. Looking back, however, I would advise teaching these commands as the students find it necessary **during** programming.

As the students got into the actual programming, wonderful things happened. They discovered for themselves how to use a counter by means of a FOR/NEXT statement when they needed to make the story pause in order to read it. Using our system, the students discovered that the CLR/HOME key would make their stories neater. One student, knowledgeable about computers, put color into his group's story by using the POKE statement. The following are some excerpts from the stories, programmed, which demonstrate these commands.

```
1570 PRINT "        YOU MUST NOW SEARCH MISS ROBINSON'S"
1580 PRINT "OFFICE FOR ANY CLUES OR EVIDENCE. AS YOU"
1590 PRINT "ARE SEARCHING, YOU FIND A MEMO FROM"
1600 PRINT "THE DAY THAT SHE WAS MURDERED. IT"
1610 PRINT "SAYS: RUSSIAN SPY IN WHITEHO—."
1620 PRINT "YOU DECIDE THAT SHE WAS MURDERED"
1630 PRINT "WHILE WRITING THIS BECAUSE IT WAS"
1640 PRINT "WRITTEN IN PEN, NEVER FINISHED, AND"
1650 PRINT "WHEN THEY FOUND HER SHE WAS HOLDING"
1660 PRINT "A PEN."
1665 PRINT "FOR X= TO 15000: NEXT X
1670 PRINT "(shift clr/home)"
```

```
900 PRINT "(shift clr/home)":POKE 53280,0:POKE 53281,1
1000 PRINT "       *THE QUEST*"
1010 PRINT "*FOR THE DIAMOND OF LIFE*"
1020 PRINT
1030 PRINT "     IT IS FOUR MOONS PAST SUNDOWN."
1040 PRINT "YOUR VILLIAGE HAS JUST BEEN ATTACKED"
1050 PRINT "BY CLUSTERS OF DRAGONS LED BY THE MOST"
1060 PRINT "VICIOUS LORD OF DRAGONS, TIAMAX, THE ONE"
1070 PRINT "YOU MUST DESTROY!"
```

The following program illustrates how the INPUT statement was used to allow the reader to **Choose His Own Adventure.**

```
2020 PRINT "     BULLETS GO FLYING EVERYWHERE. YOU"
2030 PRINT "GET SHOT IN THE FOREHEAD. BUT, THANKS TO"
2040 PRINT "THE IRON PLATE IN THE MIDDLE OF YOUR"
```

```
2050 PRINT "FOREHEAD THAT YOU GOT IN VIETNAM, IT"
2060 PRINT "RICOCHETS OFF AND ONLY SKINS YOU. YOU"
2070 PRINT "GET IN THE PLANE AND TAKE OFF UNDER ENEMY
2080 PRINT "FIRE. YOU MAKE IT SAFELY OUT OF THE HANGAR."
2090 PRINT "SOON THEY HAVE M-15 MAGNUM JETS ON YOUR TAIL."
2100 PRINT "IF YOU CHOOSE TO OUTFLY THEM, PRESS E."
2110 PRINT "IF YOU CHOOSE TO STAY AND FIGHT, PRESS F."
2120 INPUT A$
2130 IF A$="E" THEN 3000
```

After they completed programming the stories, the other groups in the class ran them on the computer, read them, and rated them on a scale frum 1 to 5, with 5 being a super exciting adventure story that was fun to read. The disks were then made available to other students in the school who use the computer lab. The sixth graders at our school enjoyed reading all the different endings.

In six weeks' time, it had become obvious to me that a great deal of learning had taken place in my seventh grade English classes. The students learned about writing, programming, and cooperation among themselves. They even got excited about narratives, and surprisingly, students wanted to spend time outside of school hours working on their stories. The most important lesson I learned is that teachers need to realize what we've been told time and time again. We need to capitalize on our students' natural interests and enthusiasm in order to get them turned on to school assignments. When this happens effectively, real learning can take place. If you chose to take this adventure, follow these tips:

1. Set guidelines about what you will accept before the students begin writing. Junior high students can have extra wild imaginations.
2. Set deadlines for all drafts.
3. Teach beforehand how the PRINT statement works.
4. Allow noise — real learning and interaction involve communication.
5. Let students try whatever they want on the computer. The worst thing that will happen is they'll erase their disk or program.

Anne Weber Dunham taught seventh grade math, reading, and English at Highland Middle School in Ogden, Utah. She is currently coaching at Northern Michigan University and plans to re-enter the classroom and expand upon the activities described in her article. For further information on her seventh grade writing program, contact:

Anne Weber Dunham
145 West Arch
Marquette, MI 49885

PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
PLAGIARISM
WILL THE COMPUTER BE TO BLAME?

by John B. Howe

Editors' note: John Howe raises an unpleasant but crucial problem in instruction in programming—students copying other students' work. As in other subjects where learning is best demonstrated through actual performance, such as writing, plagiarism has emerged as a serious problem for many teachers. Many are especially stung when they first become aware of the problem and of the difficulty of detecting its occurrence—far greater than in the case of writing. Howe's contribution is not simply to raise the issue, but also to suggest a practical, unemotional solution to it.

I found myself in front of my first computer programming class four years ago. As a teacher of mathematics, the progression was a natural one, and I found the transition easy. There were a number of things that I found to be quite different from teaching in a traditional setting. The most unusual problem I encountered developed during the first semester that I was teaching beginning BASIC.

When we were approximately one-third of the way through the one-semester course, I realized there would be a major problem in teaching computer programming. I was grading my students more heavily on their performance at the terminal, based on their submitted projects, than on a sterile, in-class examination. After all, wasn't the course designed to teach programming? What better test of their understanding of computer programming, than to ask the students to create a program to accomplish some task? The major fault in my philosophy was that nowhere in education is the seduction of the student towards plagiarism more profound than in a computer programming class. Before that, I had not picked up on the fact that students were using the same program and merely putting their names on it. The programs I had assigned to date were so simple that there were very few things you could do to vary the program, and therefore, every program **should** look alike.

We finally arrived at a point in the course where I could give them a program of some substance, requiring the use of much of what we had learned, and having a number of possible approaches. When the projects were turned in, the plagiarism was impossible to miss. Almost half of the students had copied someone else's work! I had required each student to have three REM statements at the beginning of each project; their name, the date, and title of the project. What many students had done was to call up some other student's program and change the REM statement containing the other student's name to their names. They then listed and ran the program, and submitted it as their own work. These students were so foolish as to not even change programs that used "odd" line numbers such as 13 or 243, or unusual variables such as "i" or "j."

I believe one reason for this unexpected occurrence was my lack of experience working in a lab situation. I had always taught my mathematics classes with a more traditional approach. It was now clear why some of the students were doing so poorly on the tests, even though they had completed all of the projects.

I was furious! What was even more annoying was to have to grade multiple copies of the same program. I toyed with various ways of dealing with the situation. Should I give a program a grade and divide it by the number of copies turned in by various students, and that would be the grade each student would receive for that project? No, that penalized the student who originated the program, who was sometimes difficult to identify in a particular class. I decided to approach the problem from a different angle. I did not feel that the students were to blame; the enticement for computerized plagiarism was merely too great and I would have to develop a strategy to eliminate that enticement.

Over the next two months I began stockpiling programming questions from various sources. I then typed out these questions, four to a page, so when cut apart each question would be almost the size of an index card. I finally developed a resource pile of approximately 200 questions, each coded to identify a particular topic (decision-making, strings, etc.). I xeroxed my originals, so I could use them year after year, and cut the questions up on the school's electric paper cutter. I put all of the questions in a box, and the students picked (without looking) which program they would be assigned to do. I then recorded in my grade book the program number the student selected so I could keep track of which program each student had to turn in. Now everyone had a different program to do, which eliminated the outright plagiarism of someone else's work, since no two students were doing exactly the same thing. There was still consultation between students, but then it became more of a learning experience, with one student teaching another, rather than "gimme what you got."

The work in initiating this program was tremendous, but now I just add to the repertoire of questions I have already compiled. I feel that the students get much more out of this method, and I feel much more comfortable grading programs that the students have worked on independently. When the students leave my class now, they have a much better knowledge of the BASIC language.

I have discussed this plan of action with many computer teachers, and they seem to have one main concern: "How do **you** (as the teacher) know how to write all of these programs? Where do you find the time to write the programs so you know how they should be done?" The answer to that is that initially I don't do anything except assign the problem. I feel that a major part of teaching computer programming is for the students to learn the thought processes necessary to write a program. That includes writing what they think will work, trying it on the equipment, and then working out the bugs. I am not interested in "cookbook learning"—do it this way all of the time. The **primary** question to answer when looking at the work of a first-time computer programmer is: "Does the program perform the task required completely and successfully?" If it does, then the student has come a long way in understanding programming. Will you get "spaghetti programs," with multiple GOTO's? Of course! But once a student has begun to understand the principles of programming, then you can concentrate on the finer aspects of top-down programming.

[Th]e problem of plagiarism is sure to grow as we find more home computers in use. It probably will not be long before students realize that when one student turns in a report for Mr. Jones on the Civil War, his best friend need only borrow the disk, make a few name changes, and that same report, in less than five minutes, has been typed and is also ready to be turned in to another social studies teacher. Teachers are going to have to find new and better ways to prevent such abuses of today's technology, while at the same time developing a positive approach to the use of this technology to improve the quality of classroom instruction.

John B. Howe teaches mathematics at Smithtown High School West in Smithtown, New York. He has taught word processing as well as beginning and advanced BASIC, *Pascal, and computer graphics. Some of his future goals focus on the effective use of computer tools in educational administration. For further information on his program, contact:*

John B. Howe
420 Spruce Avenue
West Islip, NY 11795

FROM THE INTRODUCTION

Stories to think with, Seymour Papert might call them, these articles by teachers about their experiences using computers in their classrooms. Not theories or research reports, though we have nothing against theory and plead guilty to committing research from time to time, ourselves. Not even software reviews. First-hand accounts of what teachers have done with kids and computers in the classroom, how kids responded, and what and how teachers have learned from the experiences.

Charles L. Thompson is Co-Director of the Educational Technology Center at the Harvard Graduate School of Education. ETC was created by the National Institute of Education in 1983 to research ways of using information technologies to improve education in science, mathematics, and computing at the elementary and secondary levels. Previously Dr. Thompson directed the learning technology unit at Education Development Center in Newton, Massachusetts. At EDC, he worked with Judah Schwartz to initiate a mathematics software series—including **SemCalc** and **The Geometric Supposer**—published by Sunburst Communications. Dr. Thompson's principal research interests are in the process of educational change, including the introduction of computers and other information technologies into schools and classrooms.

Larry Vaughan is Manager for Research and Technical Assistance at the Northeast Regional Exchange, Inc. He has been involved with computer instruction for over fifteen years. During the past five years Mr. Vaughan has concentrated on helping schools make effective use of instructional technology in curriculum areas. He has edited a number of resource books for educators including **Technology Programs That Work** and **Evaluation of Educational Software: A Guide to Guides.** In 1985 Mr. Vaughan was the executive producer of "Computers in Education: A Mid-80's View," a series of three 90-minute video programs broadcast via satellite to a nationwide audience. Also, he developed an electronic newsletter for educators on The SOURCE called "TECH TALK." His most recent work has focused on promoting the use of tool software within the existing curriculum.